AD-A285 430

COMBAT RATION ADVANCED MANUFACTURING TECHNOLOGY DEMONSTRATION (CRAMTD)

"Design and Development of a CIM Architecture for Food Manufacturing"

Short Term Project (STP) #4

FINAL TECHNICAL REPORT
Results and Accomplishments (October 1989 through January 1993)
Report No. CRAMTD STP #4 - FTR 6.0
CDRL Sequence A004
July 1994

CRAMTD CONTRACT NO. DLA900-88-D-0383
CLIN 0003

Sponsored by:
DEFENSE LOGISTICS AGENCY
Cameron Station
Alexandria, VA 22304-6145

Contractor:
Rutgers, The State University of New Jersey
THE CENTER FOR ADVANCED FOOD TECHNOLOGY*
Cook College
N.J. Agricultural Experiment Station
New Brunswick, New Jersey 08903

Principal Investigator: Thomas Boucher
Co-principal Investigator: Mohsen A. Jafari

Dr. John F. Coburn
Program Director

TEL: 908-445-6132
FAX: 908-445-6145

*A New Jersey Commission on Science and Technology Center

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
JULY 1994

3. REPORT TYPE AND DATES COVERED
FINAL Oct. 1989 -- Jan. 1993

4. TITLE AND SUBTITLE
Design and Development of a CIM Architecture for Food Manufacturing (Short Term Project -- STP#4)

5. FUNDING NUMBERS
C-DLA900-88D-0383
PE - 7811s
PR - 88003

6. AUTHOR(S)
Thomas Boucher and Mohsen A. Jafari

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rutgers, The State University of New Jersey
The Center for Advanced Food Technology
Cook College, NJ Agricultural Experiment Station
New Brunswick, NJ 08903

8. PERFORMING ORGANIZATION REPORT NUMBER
FTR 6.0

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Logistics Agency
Cameron Station
Alexandria, VA 22304-6100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In collaboration with two Combat Ration Producers, one of MRE pouches and the other of tray pack, a detailed study was made of the functional and informational requirements to operate a food manufacturing plant. These inputs were used to develop formal models, "Architectures", of the functional requirements and the informational requirements. Although particular attention was given to the operating practices of combat ration manufacturers, practices of civilian product manufacture are also specified within the Functional and Informational Architectures. Two case studies were included: MRE Pouch - Omelet with Ham and Tray-Pack - Beef Chunks and Gravy. The implementation of Computer Integrated Manufacturing requires the development of a factory database that supports operating the manufacturing enterprise. A single-user Oracle Database management system was purchased and a preliminary physical database model was constructed as a prototype. Based on the architectures developed, the installed database management software, and the preliminary database modules, it was demonstrated that the proposed Computer Integrated Manufacturing of Combat Rations was achievable and would be practical for commercial utilization.

14. SUBJECT TERMS
Automation, Process, Packaging, Rations, Tray-Pack Ration, MRE (Meal, Ready-to-Eat)

15. NUMBER OF PAGES
209

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

"Design and Development of a CIM Architecture for Food Manufacturing"
Short Term Project (STP) #4

Contributing Technical Personnel

Principal Investigator	Dr. Thomas Boucher
Co-principal Investigator	Dr. Mohsen A. Jafari
Faculty	Dr. Nabil R. Adam Dr. Sooyoung Kim
Graduate Students	T. Chamberlin A. Gangopadhyay R. Holowczak J. McPhail A. Sevinc V. Srinivasan J. Weber
Consultant	Dr. E. Goldman

1.0. Results and Accomplishments

1.1 Introduction and Background

Short Term Project #4, "Design and Development of a CIM Architecture for Food Manufacturing", was to develop a Functional Architecture, an Informational Architecture, and a preliminary Database Design for packaged food manufacture. This activity was deemed necessary to pursue one of the goals of the CRAMTD program, which is to demonstrate Computer Integrated Manufacturing in the manufacture of Combat Rations. A secondary objective was to develop computer simulation models of automated tray pack and MRE pouch production lines. This activity was deemed necessary to evaluate the performance of proposed designs and to compare them to current base line practices.

In collaboration with two coalition companies, one a producer of MRE pouches and the other a producer of tray packs, a detailed study was made of the functional and informational requirements to operate a food manufacturing plant. These imputs were used to develop a formal model of the functional requirements and a formal model of the informational requirements. These models were documented in Technical Working Paper, TWP #37, "Functional Architecture for Packaged Food Manufacture" and TWP #52, "Informational Architecture for Packaged Food Manufacture", available from the Rutgers University (CAFT) Center for Advanced Food Technology . All technical working papers subsequently cited in this report are available from CAFT.

The implementation of computer integrated manufacturing requires the development of a factory database to implement the informational and data requirements that support the activities, or functions, for operating the manufacturing enterprise. A single user Oracle Database management system was purchased and a preliminary physical database model was constructed as a prototype for supporting the CIM design. The results of this work was documented in a Technical Working Paper #56, "Preliminary Database Design for CRAMTD Demonstration Plant."

STP#4 began in October, 1989 based on the proposal submitted to the DLA on August 1, 1989 and revised on August 29, 1989 after detailed review with the DLA.

1.2 Results and Conclusions

The Functional Architecture developed in this Short Term Project (STP) is a generic architecture and can therefore be used as a reference model by any company in the packaged food industry that wishes to undertake a CIM implementation project. Particular attention was given to the operating practices of combat ration manufacturers, however, practices of civilian product manufacture are also specified within the architecture. Two case studies were included: MRE Pouch - Omelet with Ham and Tray Pack - Beef Chunks and Gravy.

The Informational Architecture model developed in Phase III is traceable to the model described in the Functional Architecture, Phase II, and visa-versa. Each of the entities depicted in the Informational Architecture is represented by a table in the Database implementation. The ORACLE database management software was selected, acquired and installed for the Database implementation and a preliminary set of forms and reports were developed to demonstrate the functionality.

During the no-cost extension of this STP, an Equipment Maintenance Module and a Quality Assurance Module were developed. These Modules consist of the necessary forms for data entry and data retrieval.

1.3 Recommendations

Based on the architecture developed, the installed database management software, and the preliminary database modules, it was demonstrated that the proposed Computer Integrated Manufacturing of Combat Rations was achievable and would be practical for commercial utilization. During the last Phase of this STP, researchers worked closely with

those of STP #16 "Implementation of Integrated Manufacturing" to transfer the technology developed. The technical recommendations for STP #16 database structure, forms and reports (including both software and hardware) have already been incorporated into that successor activity.

An important research issue which arose during STP #4 is the lack of a mechanism for the system designer to test system performance during the CIM design process without separately building a simulation model. Since the IDEF methodology, employed in STP #4, can be used to define the specification of the manufacturing system, it should also be possible to derive the specification of controllers directly from the IDEF model. This capability would also simplify and enhance technology transfer to potential industrial users. A methodology has been defined in concept that should take the IDEF0 specification of the manufacturing system, automatically generate a dynamic model that can be used to analyze the system performance (IDEF2), and then automatically generate the computer code to run the system. It is recommended, therefore, that a new STP be defined to demonstrate the feasibility of integrating the IDEF methodology with testing systems dynamics, and delivery order (DO 0017) has been issued to cover STP #24 -- "Integration of IDEF methodology with Testing System Dynamics".

2.0 Program Management

Work on this Short Term Project began in October, 1989 with Interim Funding. Project Definitization occurred on April 2, 1990. There are four phases to this STP#4 "Design and Development of a CIM Architecture for Food Manufacturing". These Phases cover:

Phase I Methodological Review
Phase II Functional Architecture
Phase III Informational Architecture
Phase IV Simulation and Database Design

Each phase was carried out as a distinct activity, but phases were allowed to overlap.

However, the linkage between the functional architecture, informational architecture, and database design required some precedence structure since each succeeding step required some information from a preceeding stage. The control of project phases was further complicated by the need to carry out studies in two different companies in order to obtain a CIM architecture that would include MRE pouch and tray pack products as well as civilian products.

2.1 Summary of Progress

- Simon Simulation Models with CINEMA graphics animation were developed for the CRAMTD tray pack and the MRE pouch lines and a technical report documenting these models was issued (TWP#33).
- A Functional Architecture for packaged Food Manufacturing was developed that covers the manufacture of MRE pouches, tray pack products, and shelf stable civilian food products. A technical report documenting this model was issued (TWP#37).
- An Informational Architecture for packaged food manufacturing was developed that covers the manufacture of MRE pouches, tray pack products, and shelf stable civilian food products. A technical report documenting this model was issued (TWP#52).
- During a no cost extension period, from August, 1992 through January, 1993, a module was added to the information architecture in the area of quality control. Appendix 4.8 documenting this module is attached.
- During a no cost extension period, from August, 1992 through January, 1993, a module was added to the information architecture in the area of machine maintenance. Appendix 4.9 documenting that module is attached.
- A single user Oracle Database management system was procured and a preliminary CIM database design was implemented based on the previously defined Functional and Informational architectures. A technical report documenting this design, including Oracle Forms and reports was issued (TWP#56).
- During the no cost extension period modules were added to the preliminary database to support the areas of quality control and machine maintenance.

- During the no cost extension period a multi-user Oracle Database management system was purchased along with a Novell netware server. The installation of this software began under STP#4. After January 1993, the installations continued under STP#16.

3.0 Short Term Project Activities

3.1 STP Phase I Tasks

Phase I is a methodological review, the objective of which is to determine the methodologies available for designing a CIM Architecture. It consists of three tasks:

Review Architecture Methodologies

Review software Development Tools

Install Development Workstation

3.1.1. Review Architecture Methodologies (3.4.1)

A thorough methodological review was undertaken. We examined the IDEF methodology of the U.S. Airforce, the hierarchial control architecture of the National Institute of Standards and Technology (NIST), the Petri net control architecture, and related work being done at other universities. This review was done through literature and conversations with individuals involved. The full report on this task, "Review of CIM Architecture Methodologies," was published as Technical Working Paper (TWP) #7.

The following are the main conclusions of this phase:

1. The IDEF methodology is the most fully developed methodology currently available in the public domain. It is the methodology of choice for this project.

2. None of the methodologies reviewed showed how the functional and informational architectures are related to the physical (communication) architecture. This issue was raised

in the NIST literature and we had to plan to address communication arc' cture separately in our work.

3. IDEF methodology, by itself, lacks the capability of analyzing operational control in shop floor and lower levels. Although IDEF2 is intended to serve this purpose, it is very poorly defined at this time. The IDEF methodology would have to be combined with Petri net methodology in order to convert IDEF to a formal controller specification that could be analyzed at the shop floor level.

3.1.2 Review Software Development Tools (3.4.2)

In order to document the IDEF models and to be able to provide professional quality drawings of these models, we evaluated commercially available software tools. After evaluating these tools, two were selected.

The AIØ software of Knowledge Based Systems, Inc. was chosen to document IDEFØ. Besides its relatively low cost, it has the capability of automatically generating a drawing from simple user inputs, such as text or keystrokes. Other software we evaluated required the user to produce the drawing in a CAD-like environment, which we considered to be more tedious.

The Model Pro software of D. Appleton, Inc. was chosen to document IDEF1X. As in the case of AIØ, the drawings are automatically generated from text or keystroke inputs.

3.1.3 Install Development Workstation (3.4.3)

Two computers were installed in an Industrial Engineering Laboratory to be used in documenting IDEF models. The AIØ software of Knowledge Based Systems was installed on an IBM AT to support functional modeling. An IBM PS2 MOD 70 was purchased to be used to support the IDEF1X Model Pro Software of D. Appleton, Inc., which requires a VGA monitor. This task, completed in the quarter ending April, 1990, completed Phase I of STP#4.

3.2 STP Phase II Tasks

Phase II of STP#4 is the Functional Architecture design, the objective of which is to design a functional architecture that would include a specification for MRE pouch, tray pack

and shelf stable civilian products. It consists of four tasks:

Review Industrial Practices

Build and Document Functional Model

Install in CRAMTD Site

Provide Technical Report

3.2.1 Review Industrial Practices (3.5.1)

This activity began on April 2, 1990, with a trip to Pillsbury/Green Giant, a coalition company that collaborated with CRAMTD personnel on the MRE Pouch Functional and Informational Models. CRAMTD investigators made three visits to Pillsbury/Green Giant during June, 1990. We interacted with company managers, administrators, and production personnel and gained a detailed understanding of the practices involved in operating a packaged food enterprise. This task was deemed completed in the quarter ending October, 1990.

3.2.2 Build and Document Functional Model (3.5.2)

After the interviews in June of 1990, CRAMTD investigators began to document the Functional Architecture for the MRE pouch processes. The documentation was done in stages. The process of developing the functional model consisted of on-site interviews concerning a selected subset of functions, followed by documenting the results of the interviews in the formal IDEFØ modeling language. This would be followed by a return trip to the coalition company to review the document, make any needed corrections, and continue the interview process. Toward the end of June, 1990 and through the Fall of 1990, this process continued. By January, 1991, a functional model that could support the MRE pouch requirements was complete.

During the spring of 1991 we initiated a collaboration with Venice Maid Food Company to extend the Functional Architecture to tray pack and shelf stable civilian products. The model development process for tray pack and civilian products continued through the summer of 1991 and the combined architecture, including MRE pouch, was essentially complete by the Fall of 1991.

3.2.3 Install in CRAMTD Site (3.5.3)

In Fall, 1991, computer disks incorporating the Functional model were transferred to CRAMTD management. Subsequently, the architecture was presented at the Annual Contract Briefing in October 1991.

3.2.4 Technical Report (3.5.4)

In December, 1991, the document entitled "Functional Architecture for Packaged Food Manufacturing" (TWP37) was issued as the final technical report for Phase II. It describes a "Generic Architecture", which characterizes both civilian and military manufacturing environment. There are also two appendices to that report, which are case studies of the application of the generic architecture. One case study is for the manufacture of a MRE pouch product; the other case study is for the manufacture of a tray pack product.

Phase II of STP #4 was reported complete in the quarterly report ending January, 1992.

3.3 STP Phase III Tasks

Phase III of STP #4 is the Informational Architecture design, the objective of which is to design an informational architecture that would include a specification for MRE pouch, tray pack, and shelf stable civilian products. It consists of four tasks:

Review Industrial Practices

Build and Document Informational Model

Install in CRAMTD Site

Technical Report

3.3.1 Review Industrial Practices (3.6.1)

This activity began coincident with the documenting of the functional architecture in June, 1990. This led to some natural overlapping between phases II and III of this STP. The research team continued on-site observations of data collection practices in packaged food manufacturing environment through April, 1991.

3.3.2 Build and Document Informational Model (3.6.2)

This task began in the summer of 1990. At that time we addressed the MRE pouch manufacturing environment. This activity continued through the spring of 1991. During the

spring of 1991 and through the fall of 1991, the research team addressed the informational requirements of the tray pack and civilian products.

The process of collecting data for the information model involved gathering the forms and reports currently being used to maintain data by the food companies that collaborated with us. These data elements were then reviewed to eliminate redundancy and to prune the data set. Using the IDEF1X methodology, data entities and attributes were defined based on these collected forms and reports. This activity was essentially completed in the Fall of 1991.

3.3.3 Install in CRAMTD Site (3.6.3)

Computer disks of the Informational model were transferred to CRAMTD management.

3.3.4 Technical Report (3.6.4)

In April 1992, the document entitled "Informational Architecture for Packaged Food Manufacturing" (TWP52) was issued as the final technical report for Phase III. This included an informational architecture that covered both civilian and military product manufacture. This architecture, defined in IDEF1X modeling language, became the basis for the preliminary database design.

3.4 STP Phase IV Tasks

Phase IV of STP#4 includes the design and development of computer simulation models and the implementation of a prototype CIM database. Phase IV consists of five tasks:

Design and Code Simulation Model

Install in CRAMTD Site

Technical Report (Simulation)

Design Preliminary Database

Technical Report (Database Design)

3.4.1 Design and Code Simulation Model (3.7.1)

The design of the simulation models began in June, 1990 and a software requirements specification, "Simulation Model, Software Requirements Specification, Version 1.0", was released in October, 1990 as a Technical Working Paper (TWP15). The development work was done using Simon simulation language and cinema graphical display. The tray pack simulation model was completed in the Fall of 1990, and the pouch simulation model was completed in June, 1991.

3.4.2 Install in CRAMTD Site (3.7.2)

In June, 1991, the simulation programs were demonstrated to CRAMTD management. Run time files were transferred.

3.4.3 Technical Report (Simulation) (3.7.3)

In Fall, 1991, the document entitled "Report on CRAMTD Tray Pack and MRE Pouch Simulation Models" (TWP 33) was issued as the Final Technical Report. It contains the Simon and Cimema programming codes as well as outputs for test simulation runs.

3.4.4 Design Preliminary Database (3.7.4)

In March, 1991, a single user Oracle Database Management System was procured and in April, 1991, the research team began implementing a preliminary database design based on the informational architecture developed for an MRE Pouch Enterprise. As the data requirements for tray pack and civilian enterprise was developed in the Fall of 1991, we began to incorporate these requirements into the prototype database.

In August, 1992, STP#4 was granted a no cost extension through January, 1993. During this period, preliminary database modules were developed for raw material quality control, finished goods quality control, and machine maintenance.

3.4.5 Technical Report (Database Design) (3.7.5)

In July, 1992, the document entitled "Preliminary Database Design for the CRAMTD Demonstration Plant", (TWP56) was issued as a Final Technical Report. It describes the relational database model and the Oracle Database Management System. The database tables created are related to the IDEF1X Informational Model for completeness in understanding the transition from data model to database Implementation. Structured Query Language

(SQL) Forms and SQL Reports are described and user screens are shown as the interface between the database and the functions it supports. During the development of the preliminary database, a Technology Transfer partner was identified for the commercialization of the final database to be designed and implemented under CRAMTD Phase II.

4.0 Appendix

4.1 Figure: "Time Events and Milestones"

4.2 "Review of CIM Architecture Methodologies", TWP#7.

4.3 "Functional Architecture for Packaged Food Manufacture", TWP#37.

4.4 "Informational Architecture for Packaged Food Manufacture", TWP#52.

4.5 "Simulation Model, Software Requirements specification, Version 1.0", TWP#15.

4.6 "Report on CRAMTD Tray Pack and MRE Pouch Simulation Models", TWP#33.

4.7 "Preliminary Database Design for the CRAMTD Demonstration Plant", TWP#56.

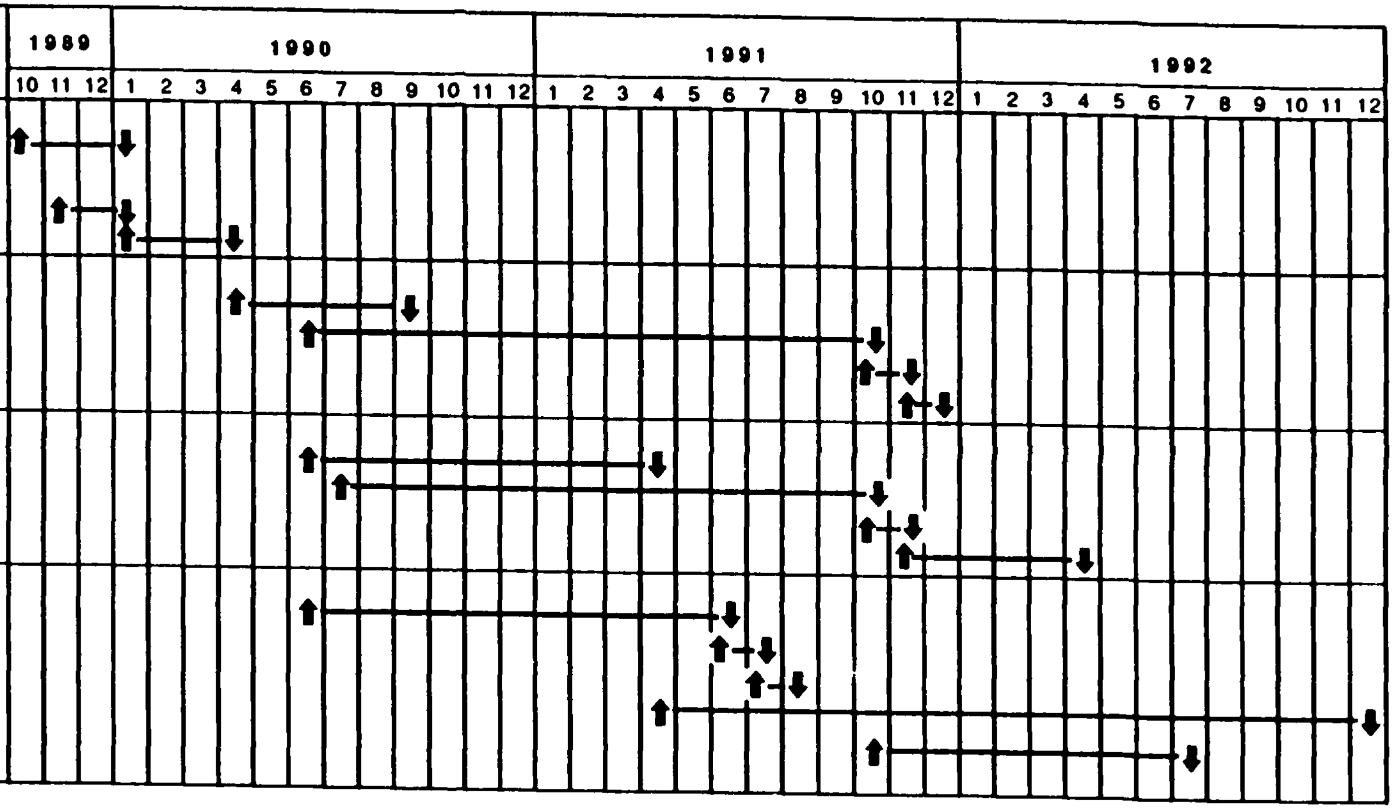
4.8 "Report on Quality Assurance Module Implementation -- Part I".

4.9 "Report on Equipment Maintenance Module Implementation -- Part I".

Appendix 4.1

Fig. 1 - CRAMTD Short Term Project #4
Design and Development of a CIM Architecture
for Food Processing
Projected Time & Events and Milestones

Phase	Description of Specific Task	Section Ref. #
I	Methodological Review	3.4
	Review Architecture Methodologies	3.4.1
	Review Software Dev'mt Tools	3.4.2
	Install Development Workstation	3.4.3
II	Functional Architecture	3.5
	Review Industrial Practices	3.5.1
	Build and document funct'l model	3.5.2
	Install in CRAMTD site	3.5.3
	Technical Report	3.5.4
III	Informational Architecture	3.6
	Review Industrial Practices	3.6.1
	Build and document inform. model	3.6.2
	Install in the CRAMTD	3.6.3
	Technical Report	3.6.4
IV	Simulation and Data Base Design	3.7
	Design and code Simulation Model	3.7.1
	Install in CRAMTD site	3.7.2
	Technical Report (simulation)	3.7.3
	Design Preliminary Data Base	3.7.4
	Technical Report (Data Base Design)	3.7.5



Legend: ⬆ - start ⇧ - projected start ⬇ - completed ⇩ - projected completion

Revised 04/26/93

COMBAT RATION ADVANCED MANUFACTURING TECHNOLOGY DEMONSTRATION (CRAMTD)

Review of CIM Architecture Methodologies (Ref. No. 3.4.1)

Technical Working Paper (TWP) 7

T.O. Boucher and M.A. Jafari
Department of Industrial Engineering
Rutgers University

CRAMTD CONTRACT NO. DLA900-88-D-0383
CLIN 0003

Sponsored by:
DEFENSE LOGISTICS AGENCY
Cameron Station
Alexandria, VA 22304-6145

Contractor:
Rutgers, The State University of New Jersey
THE CENTER FOR ADVANCED FOOD TECHNOLOGY*
Cook College
N.J. Agricultural Experiment Station
New Brunswick, New Jersey 08903

DR. JACK L. ROSSEN
Program Director, CRAMTD
DR. JOHN F. COBURN
Associate Director, CRAMTD

TEL: 908-932-8306
FAX: 908-932-8690

*A NEW JERSEY COMMISSION ON SCIENCE AND TECHNOLOGY CENTER

1. Introduction

The manufacturing enterprise is a complex set of relationships of humans, material, machines, tools, and information. Attempts to use the computer to assist in the manufacturing function has often proceeded in a disjoint fashion, resulting in a patchwork of hardware and software that addresses sub-problems in a suboptimal fashion. For example, factory accounting systems, one of the earliest sub-systems to be computerized, are often not designed with information that can be usefully shared by production functions. Material requirements planning (MRP) software, the function of which is to provide overall production planning and control, is usually not integrated with shop floor control systems.

The fundamental problem is that of interrelating data and knowledge across the manufacturing organization. Computer Integrated Manufacturing (CIM) is an approach to accomplish that goal through computer data bases and their administration which integrate manufacturing functions (such as process control and quality control) and business functions (such as purchasing, accounting, and inventory control). The principal benefits of this integration are the elimination of redundancy, the increased speed of performance, the reduction in non-value added functions, and the improved accuracy and quality of data.

Organizations are typically described by an organization chart, which shows the hierarchical structure of the organization. Departments are described by functional names, but the organizational chart does not show functional relationships.

Understanding organizational dynamics requires a different view. An organization moves toward achieving its goals through the performance of functions, which are highly interrelated. A functional view of an organization requires that the inputs and outputs of each organizational activity be modeled, including relationships and dependencies. An organizational function or activity may or may not correspond to the organizational chart, which simply shows the hierarchy and reporting relationships. The organizational chart is not necessarily relevant to the functions and activities of the organization. It is the goal-oriented functions and activities that represent the organizational dynamics.

The performance of functions usually requires one or more of the following: material, tooling, machines, or information. Every function has a world view associated with it, which includes a view of the information, which includes data, and the format of the information that is necessary to accomplish that function. We shall call this the "user view".

The function may both use data and create data. The data created by a particular function may be part of the user view of another function. One of the ways in which functions are interconnected is through the information which they share or which one function provides to another.

In order to realize the full benefits of CIM, it is necessary to understand the interrelationship of functions with each other and their relationship to the information resources of the organization. It is also necessary to understand the structure of the information resources: what they are, how they are used, how they are changed.

The term "Architecture" has been used to describe a set of descriptive models of the structure of the relationship among entities in an organization. Architecture is a generic term and there is more than one approach to modeling the organization architecture as a necessary first step toward CIM. It is the purpose of this report to review the approaches that are currently available in the public domain. The principle approaches reviewed are the IDEF methodology of the U.S. Air Force, the architecture of the National Institute of Standards and Technology, and the Petri Net architecture approach. Each will be described in a separate section, followed by a brief description of other methodologies. The reader is supplied with an extensive bibliography that goes into more detail on each of these subjects.

2. IDEF Methodology

Major software development projects have always proved extremely difficult to manage. The success of these projects depend critically on how well the overall problem is defined before programming begins. In order to improve that definition, a number of structural design methodologies were developed in the 1960's and 1970's to assist systems analysts in designing information systems.

In 1973 a more general systems engineering design methodology was introduced by Douglas Ross of MIT under the name "Structured Analysis and Design Techniques" (SADT). The development of this methodology was motivated by a commercial project to design a complete factory of the future. The methodology was more specific to system design in manufacturing than earlier structured analysis methodologies.

In the late 1970's the U.S. Air Force sponsored a series of projects to design an Architecture for Computer Integrated Manufacturing. One of the Architectures was to be an activity only, or functional architecture. It was developed using the SADT approach and has become known as the IDEFϕ methodology. This was followed by the development of a composite information modeling methodology, which became known as IDEF1. In 1985, an extended version of IDEF1 (IDEF1X) was released. It provided improvements to the graphical representation and modeling procedures and enhancements to the semantics of IDEF1.

2.1 IDEFϕ

The IDEF modeling procedure is used to model the activities that go into running a business. Modeling teams use the IDEF methodology to document the way in which a factory is actually operated. This is called the "As Is" model, which becomes a baseline for analyzing inconsistencies, redundancies, and non-value added activities in the organization. The "As Is" model becomes a baseline for developing a "To Be" model, which eliminates the deficiencies of the "As Is" state.

The "As Is" model is developed through a process of top down functional analysis, beginning with the overall goal of the organization. For example, Figure 1 illustrates a high level activity for a manufacturing organization, to "manufacture food product". The interfaces to a function are indicated by arrows that enter and leave the function. As Figure 1 illustrates, they are classified as inputs, outputs, controls, and mechanisms. Input arrows, which appear to the left of the box, represent things that are used and transformed by the function. An input can be raw material (in a material processing function) or information (in an information processing function). The Manufacture Food Products function requires both. The procurable items are the materials that are processed into food products. The contract schedule is information that is processed into a production plan.

Controls, which appear at the top of the function box, occur when the execution of the function is constrained by an entity outside the function. As illustrated in Figure 1, FDA and USDA requirements, as well as Mil Specs, which are exogeneous to the organization, are constraints on "manufacture food products".

Outputs of the function can be material items or information items or both. Finally, mechanisms are shown as an arrow entering at the bottom of the box. The mechanisms arrow indicates the resources by which the function is realized. Mechanisms may be used to indicate the job skill level (who) will do a particular function; it can be used to indicate tools or machines required.

The most general level of an IDEF model, called the AO concept level, decomposes into a set of submodels (sub functions) that comprise the top level model. Consider the illustration in Figure 2. The process employed in IDEF modeling is to gradually expand the detail of the organization by breaking out subfunctions that comprise the higher function. The rule of thumb is to expose detail by expanding the function into from 3 to 6 subfunctions. Figure 4 conceptually illustrates the process.

Figure 2 shows four major functions that comprise Figure 1. The inputs, outputs, and controls that were evident in Figure 1 are also shown entering and exiting Figure 2. In addition there are other

inputs, outputs, and controls created within the diagram due to the additional levels of detail which show the interrelationships between functions.

Function 4 (Produce Product) of Figure 2 is further broken down in Figure 3. Note that the inputs, outputs and controls and mechanisms, that enter "Produce Product" in Figure 2 are now entering at the boundaries of Figure 3. In this manner, there is a connectedness between diagrams at each level of the hierarchical decomposition. Once again, Figure 4 conceptually illustrates the process.

Creation of a model in IDEF is a process that requires the participation of more than one person. Besides the analyst, it is necessary to have individuals familiar with the functions of the organization to serve as information resources and reviewers of the models.

2.2. IDEF1X

IDEF1 and its extension, IDEF1X, is a methodology for modeling data entities and their relationships. An entity is represented by a box labeled by a noun, as shown in Figure 5. It may be a material thing, such as a tray menu item or a non-material thing, such as retort processing time. These entities are the types of information and data that are required to perform the functions of IDEFϕ.

There normally exists one or more instances of an entity. For example, an instance of the entity tray pack menu is "tray pack, mixed vegetables". Another instance is tray pack, beef chunks in gravy. Instances of an entity are identified by assigning the entity a "key attribute", such as the military specification number of the tray menu item.

Entities may have relationships to other entities. IDEF1X allows the models to be fairly specific about relationships. Each contract is written for one tray pack menu item; there may zero, one, or more contracts for the same menu. The relationship is shown by a line with a dot above the entity contract. By attaching a specific number to that we can indicate a fixed correspondence between the two entities. In this case it is one to one. A dot without a number can represent a cardinality of zero, one, or many. An entity that relates to zero, one, or many instances of another entity is a "parent" to that "child" entity.

The key attribute of an entity is placed at the top of the entity and enables the user to identify the specific instance of the entity. An entity may have non-key attributes when they are appropriate. Non key attributes give the user additional information about the entity.

A foreign key (FK) is used to identify the relationship between entities. For example, the foreign key "mil No." in the entity "contract" shows the relationship to the entity "tray pack menu". When IDEF1 models are used to design a data base, the foreign key indicates the relationship between files.

2.3. IDEF2

IDEF2 is a modeling procedure for describing the elements of the manufacturing system whose behavior varies over time. The methodology is based on network modeling concepts. There are two purposes of IDEF2: 1) to document a system in a way that it can be communicated to management and 2) to provide a means of analyzing dynamic performance.

The IDEF2 methodology is the least specific and least well-defined of the IDEF concepts. In theory it is a simulation of the functional and information submodels to whatever level of detail is reasonably possible. The methodology is not currently developed enough to standardize the level of modeling detail.

3. Petri Nets

Petri nets were originally introduced by Petri in 1962, as a formalism for representing causal relationships between events (or activities) taking place in a system exhibiting concurrency, asynchronism, and conflict. Unlike IDEF models, it has the capability of describing the flow of both physical and logical entities in a system.

A Petri net may be described through a set-theoretic type structure and/or a directed multigraph. The latter one is specially more desirable as it may represent a natural sequence of events and activities taking place in a system. In a Petri net graph (an example is shown in Figure 6) there are two types of nodes circles and bars (called "places" and "transitions" respectively). Places and transitions are connected via directed arcs representing input/output relationship. The places, transitions, and the input/output arcs basically describe the topology of the Petri net. The dynamics of the Petri net is governed through the firing of its transitions. A transition is enabled, thus may fire, if all of its input places are marked. A place is marked if there are a sufficient number (defined by the topology) of tokens there. The mechanism to fire transitions in a Petri net resembles the inference engine in AI, with the marked places representing the facts about the current state of the system. The transitions in the Petri net correspond to the rules in AI. As for the control strategy, one needs to augment transitions with some rules in order to decide which enabled transitions may fire.

Over the last three decades, there have been enormous extensions to the original work of Petri, both in application and theory. On the theoretical side, the emphasis has been on extending the modeling power of Petri nets and on developing a mathematical framework to analyze different properties of Petri nets. On the application side, Petri nets have been used for specification and verification of communication protocols, real time controllers, and discrete event systems, in general.

In the context of manufacturing, there have been several research projects (for example, S.E.C.O.I.A. project [4], [2] and [23]) to develop Petri net based controllers. The emphasis has been concentrated on the lower levels of CIM control hierarchy including shop floor controllers, cell controllers, and workstation controllers. At each level, the Petri net is in charge of coordinating or sequencing the tasks to be performed at that level. The Petri nets in different levels interact through token passing via the shared places. Note that, conventionally sequencing has been a task performed by programmable logic controllers.

To our knowledge, Petri nets have not been used to model higher levels in CIM control hierarchy. This is not surprising, as in these levels one is less interested in the flow and sequencing of entities and more interested in defining goals, setting plans, identifying necessary functions, and their interrelationship. Though Petri nets can be used to coordinate different functions and to represent the flow of control from one function to another, they can hardly be used to define these functions. One reason is that Petri nets are flow models, and another is that there has not been any attempt to augment Petri nets with a natural language as is done with IDEF models. Therefore, Petri net does not seem to be an appropriate formalism for the definition of functions and their interrelationship in higher levels of the CIM control hierarchy. Moreover, Petri nets may not be used to define information entities and their interrelationship. However, they may be used as a formalism for representing information flow and for coordinating the use of information by different functions.

One may observe that IDEF methodology and Petri nets are two different yet complementary methodologies. It is of interest to us to determine how and when these two could be combined.

4. NIST Architecture

Since the early 1980's, the National Institute of Standards and Technology ([17,18,19]) has been involved with a project to develop standards for automated manufacturing systems and to transfer technology to industry. To this end, the Center for Manufacturing Engineering at NIST has established an experimental test bed, the Automated Manufacturing Research Facility (AMRF). The design philosophy for the System is to exhibit a greater degree of flexibility and modularity than any other existing automated system. To meet these objectives, the following has been done:

1. The production control system has been partitioned into five levels in a hierarchical structure (see [17,18]).
2. The controllers for different levels were implementd in a distributed computer environment. This also allowed for a distributed data base management system.
3. Sensory equipment were used for feedback to the controllers in the lower levels.

Each controller is implemented as a finite state machine (FSM). State graphs or machines are a special type of Petri nets. The nodes in the graph represent the global state of the system being controlled. This is in contrast to Petri nets where nodes carry only local information. In a state graph, a change of state requires global information. In a Petri net only local information is required. This is a major advantage of Petri net over state machine.

Similar to the development work performed around Petri nets, the emphasis here is to develop formal models for sequencing of functions and for flow of information. The functional structure of the system is defined only informally. NIST has also performed a preliminary study ([19]) on the structure of a control system for CIM and have concluded that the three major functions, namely, production management functions; information management functions; and communication must be separately specified. They have not, however, used any structured model to describe the details of each of these functions.

5. Others

A group of researchers at Rensselaer Polytechnic Institute ([7,8,9]), have recently developed an architecture for information management in a manufacturing environment. This is based on the Metadatabase approach and TSER (Two-state Entity Relationship) methodology. Metadatabase describes data, knowledge, and control strategy. TSER contains constructs for building functional and operational models. There are algorithms to link the functional model to the operational model. There are also algorithms to map the operational model into a format which could be used to design a relational data base.

Compared to IDEF methodology, this architecture has some advantages and disadvantages. On the advantage side, the above architecture incorporates mapping algorithms from one model to another and finally to a data base design. These algorithms are not defined in IDEF methodology documentation. On the other hand, IDEF methodology has published a more concise specification scheme, which is not available in the above architecture. Moreover, the notion of using natural language in IDEF methodology makes it more attractive than TSER. At the same time both of these methodologies lack the mathematical theory which is present in Petri nets. Such a mathematical framework would enable one to formally analyze the behavior of the system.

There are also some other related work (for example [3]). These however, do not present any structured methodology. Therefore, we do not describe them in detail.

Bibliography

1. Appleton. D., "The Emerging Standard for CIM Data Sharing," Proceedings of IDEF User Group, Universal Technology Corp. May, 1989.
2. Boucher, T. O., M. A. Jafari, and G. A. Meredith, "Petri Net control of an Automated Manufacturing Cell," Computers and Industrial Engineering, Vol. 17, No. 1-4, 1989.
3. Biemans, F. P. and C. A. Vissers, "Reference Model for Manufacturing Planning and Control systems," Journal of Manufacturing systems. (to appear)
4. Bigou, J., M. Courvoisier, M. Demmou, H. Desclaux, C. Pascal, and R. Valette, "A Methodology of Specification and Implementation of Distributed Discrete control systems," IEEE Transactions on Industrial Electronics IE-34, 4, 1987, p. 417.
5 Davis, W. J. and A. T. Jones, "Functional approach to Designing Architectures for CIM," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 2, 1989.
6. Douglas, R. T. and K. E. Schoman, Jr., "Structured Analysis for Requirements Definition," IEEE Transactions on Software Engineering, Vol. SE-3, No. 1, 1975.
7. Hsu, C. and C. Skevington, "Integration of Data and Knowledge in Manufacturing Enterprises: A Conceptual Framework," Journal of Manufacturing Systems, Vol. 6, No. 4, 1987.
8. Hsu, C. C. Angulo, A. Perry, and L. Rattner, "A Design Method For Manufacturing Information Management," Proceedings of systems Integration conference, IEEE computer society Press, 1987.
9. Hsu, C., M. Bonziane, W. Cheung, J. Naguer, L. Rattner, and L. Yee, "A Metadata System for Information Modeling and Integration," Proceedings of Systems Integration Conference, IEEE Computer Society Press. (To appear in April, 1990)
10. Integrated Computer-Aided Manufacturing (ICAM) Function Modeling Manual (IDEFO), Materials Laboratory, Wright-Patterson Air Force Base, June, 1981.
11. Integrated Computer-Aided Manufacturing (ICAM) Architecture, Part II, Composite function Model of Manufacture Product (MFG0), Materials Laboratory, Wright-Patterson air force Base, June, 1981.
12. Integrated Computer-Aided Manufacturing (ICAM) Architecture, Part II, Composite Information Model of Manufacture Product (MFG1), Materials Laboratory, Wright-Patterson Air Force Base, June, 1981.
13. Integrated Computer-Aided Manufacturing (ICAM) Architecture, Part II, Composite Information Model of Manufacture Product (MFG1) - Part 2, Materials Laboratory, Wright-Patterson Air force Base, June, 1981.
14. Integrated Computer-Aided Manufacturing (ICAM) Dynamics Modeling Manual (IDEF2), Materials Laboratory, Wright-Patterson Air Force Base, June, 1981.
15. Information Modeling Manual IDEF1 - Extended, ICAM Project, U. S. Air Force, December, 1985.
16. Jafari, M. A., "A Design Methodology for Shop-Floor controllers Using Colored Petri Nets," Working Paper 89-112, Industrial Engineering Department, Rutgers University, 1989.
17. Jones, A. and C. McLean, "A Proposed Hierarchical Control Model for Automated Manufacturing Systems," Journal of Manufacturing systems, vol. 5, 1986.
18. Jones, A., E. Barkmeyer, and W. Davis, "Issues in the Design and Implementation of a system Architecture for Computer Integrated Manufacturing," International Journal of Computer Integrated Manufacturing, Vol. 2, No. 2, 1989.
19. Jones, A. and A. Saleh, "A Multi-Level/Multi-Layer Architecture for Intelligent shop Floor Control," National Institute of Standards and Technology, Unpublished Manuscript.

20. Joyce, R. and M. Nolan, "Interactive Knowledge Acquisition for Tomorrow's Enterprise," Proceedings of IDEF User Group, Universal Technology Corp, May, 1989.

21. Marca, D. A and C. L. McGowan, Structured Analysis and Design Technique, McGraw-Hill Book Company, 1988.

22. Martin, J. and C. McClure, Diagramming Techniques for Analysts and Programmers, Prentice-Hall Inc., Englewood Cliffs, NJ, 1985.

23. Meredith, G. A., "A Transformation From A Boolean Equation Control Specification to Petri Net," Unpublished MS Thesis, Department of Industrial Engineering, Rutgers University, 1990.

24. Zachman, J. A., "A Framework for Information Systems Architecture," Proceedings of IDEF User Group, Universal Technology Corp., May, 1989

Used At:	Author:	Date: 2/9/1990	Working	READER	DATE	Context
	Project: Traypack	Rev:	Draft			
			Recommended			
	Notes: 1 2 3 4 5 6 7 8 9 10		Publication			

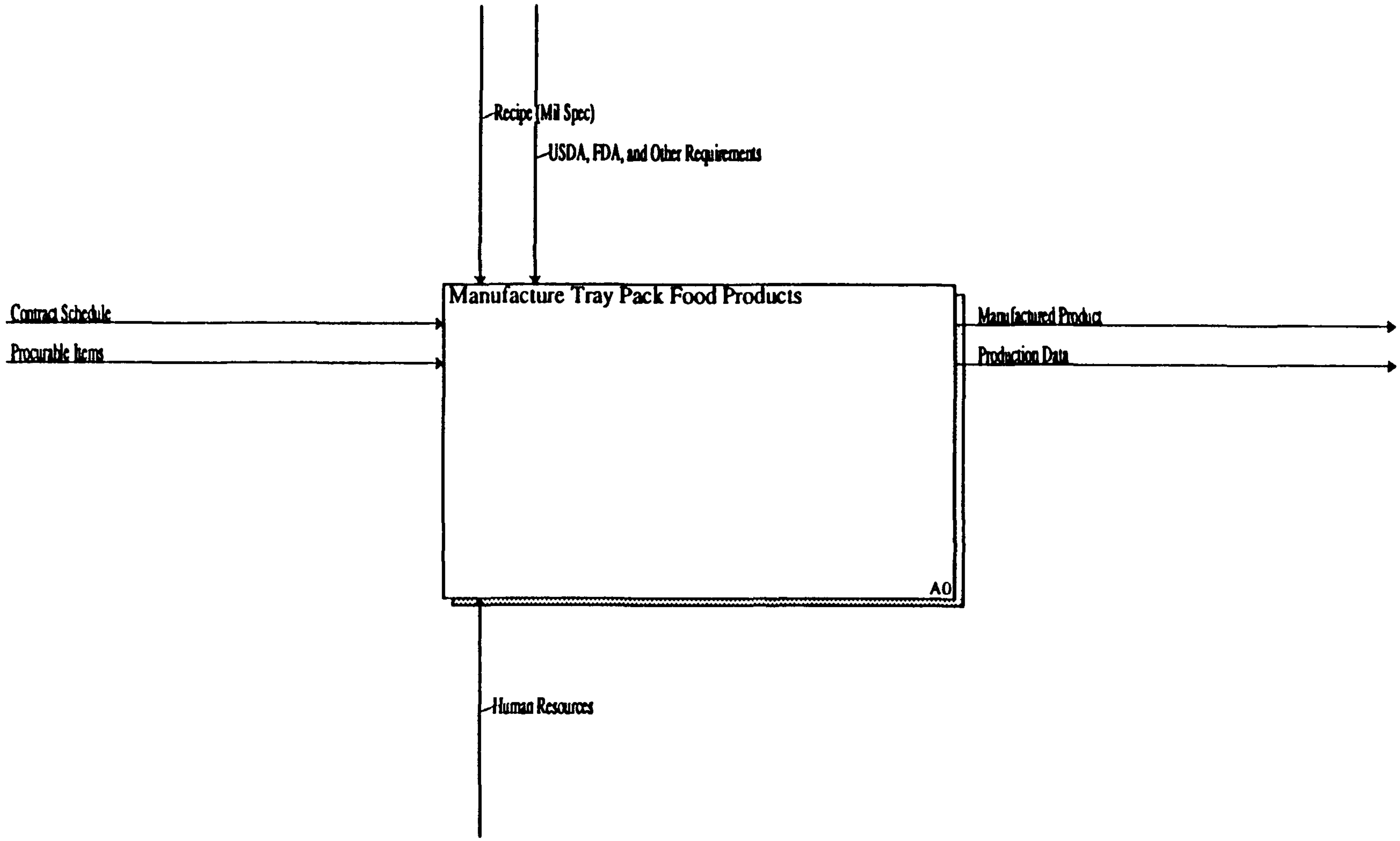
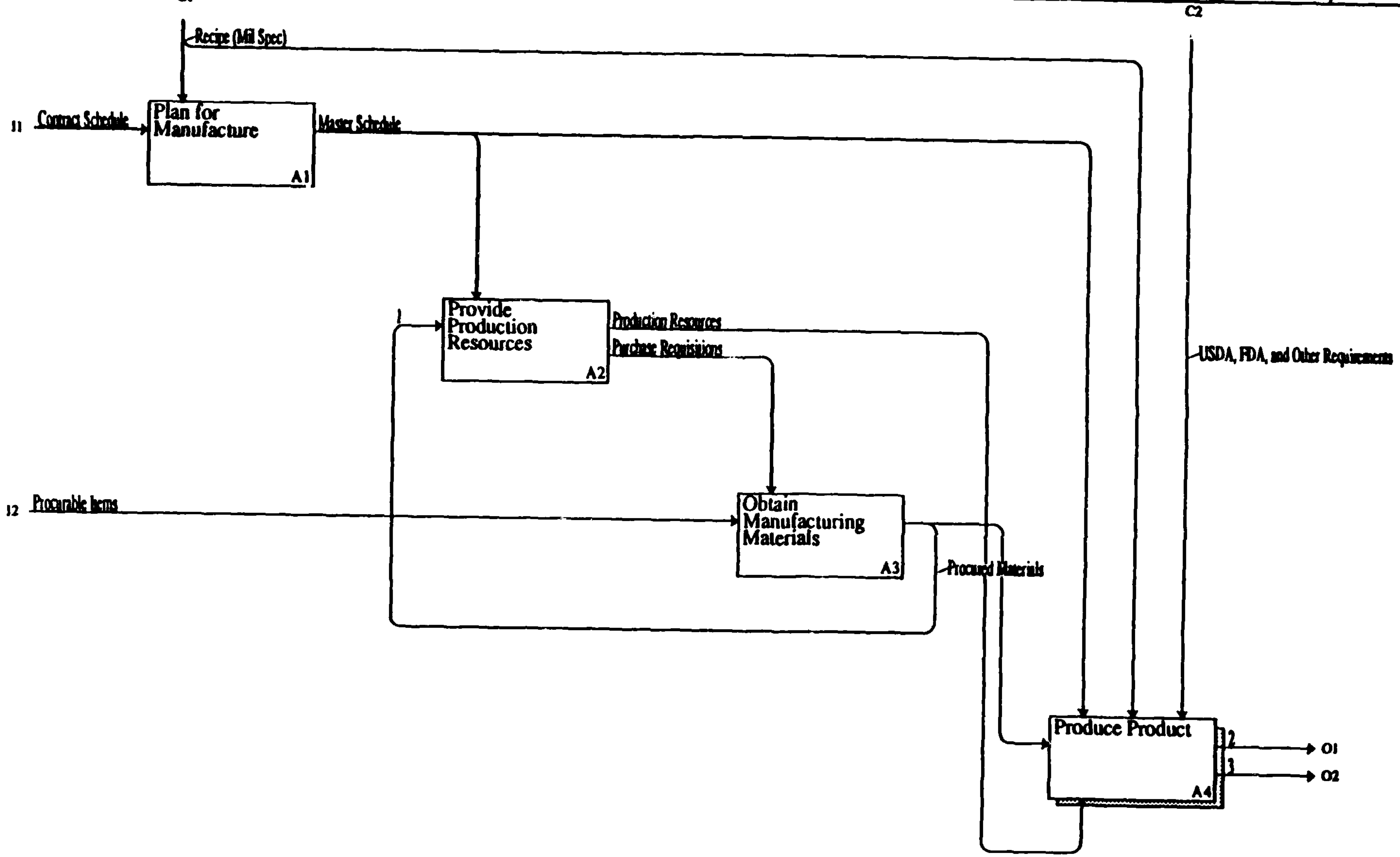


Figure 1

Node:A-0	Title:Manufacture Tray Pack Food Products (Context)	Number: 4

Used At: | Author: | Date: 2/9/1990
Project: Traypack | Rev:
Notes: 1 2 3 4 5 6 7 8 9 10

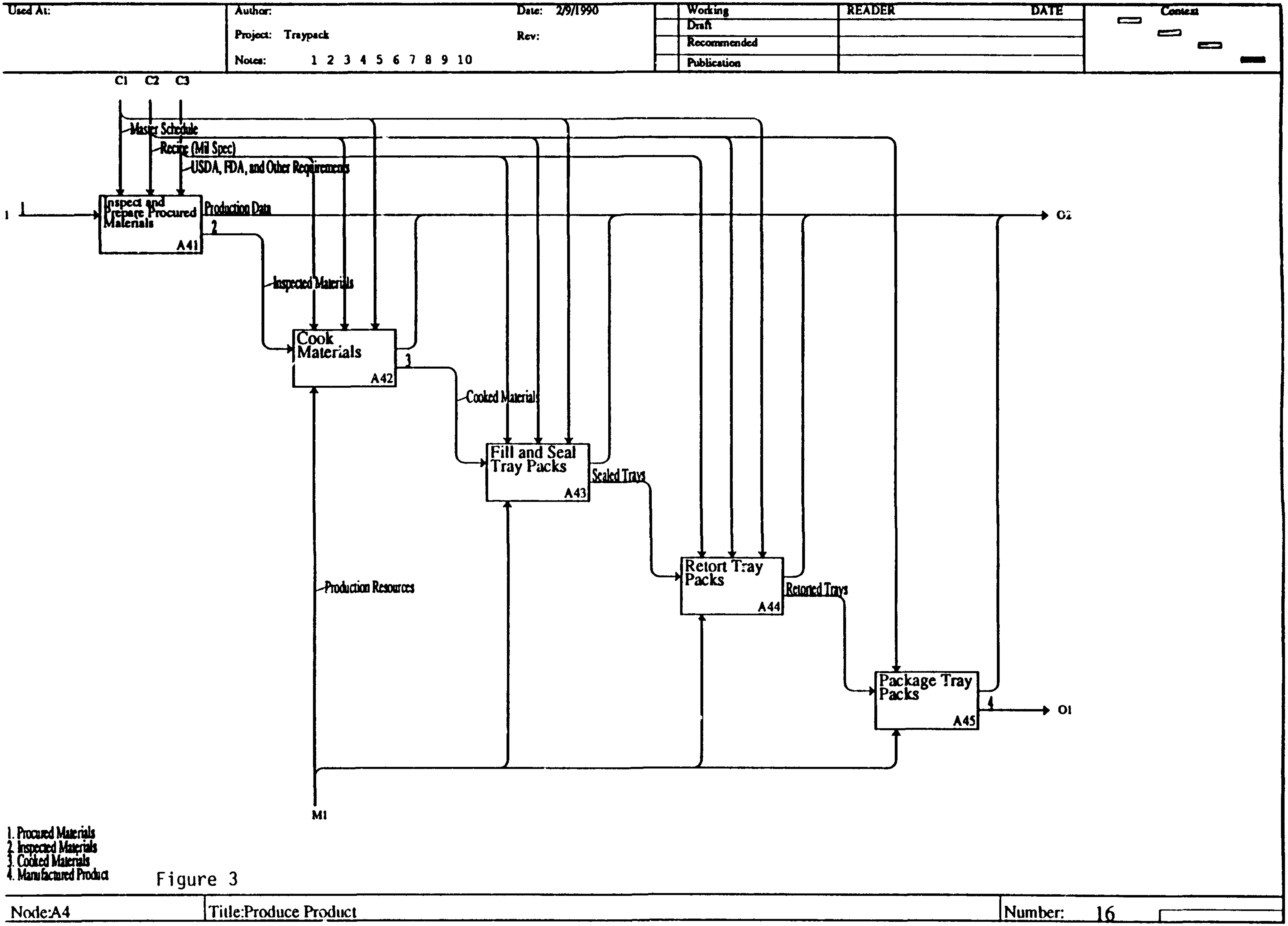
Working | READER | DATE | Context
Draft
Recommended
Publication



1. Procured Materials
2. Manufactured Product
3. Production Data

Figure 2

Node: A0 | Title: Manufacture Tray Pack Food Products | Number: 7



Used At:
Author:
Date: 2/9/1990
Project: Traypack
Rev:
Notes: 1 2 3 4 5 6 7 8 9 10
Working
Draft
Recommended
Publication
READER
DATE
Context
C1
C2
C3
Master Schedule
Recipe (Mil Spec)
USDA, FDA, and Other Requirements
Inspect and Prepare Procured Materials
A41
Production Data
Inspected Materials
Cook Materials
A42
Cooked Materials
Fill and Seal Tray Packs
A43
Sealed Trays
Retort Tray Packs
A44
Retorted Trays
Package Tray Packs
A45
O2
O1
Production Resources
M1
1. Procured Materials
2. Inspected Materials
3. Cooked Materials
4. Manufactured Product
Node:A4
Title:Produce Product
Number: 16

Figure 3

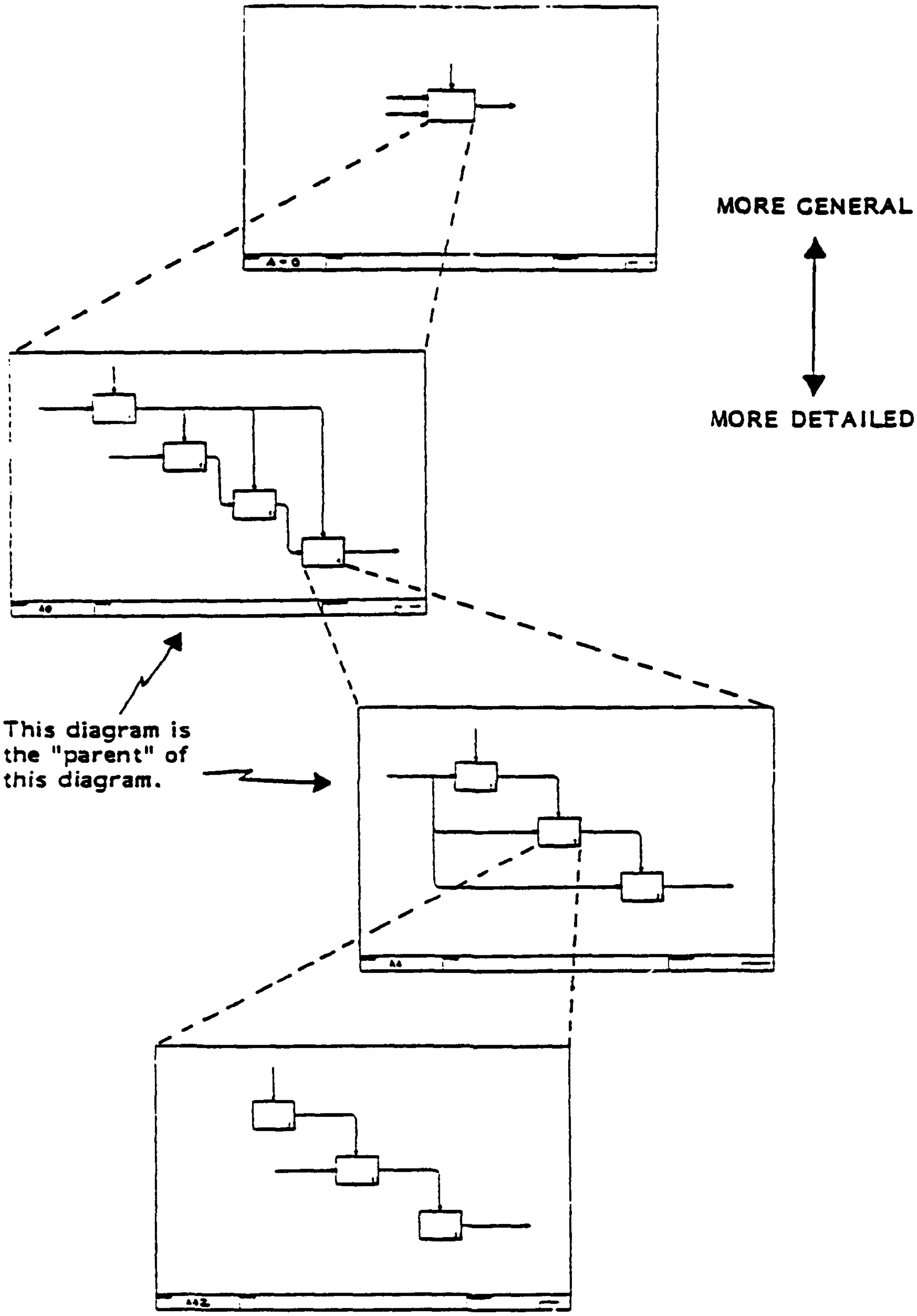


Figure 4 -- Relationship between levels in IDEFO Methodology
(Source: Ref. 15)

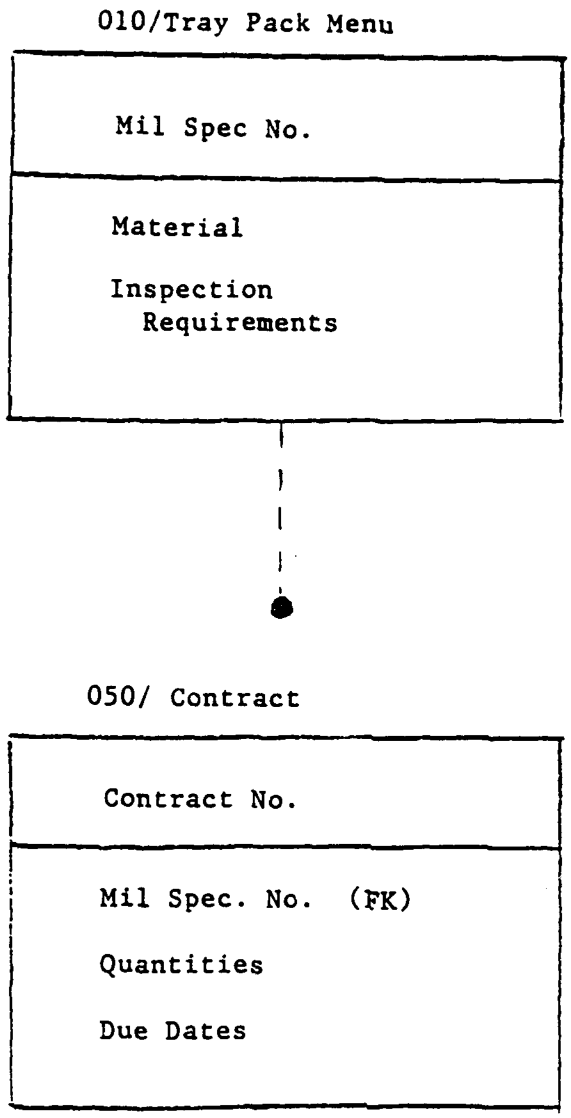


Figure 5. -- Example of Partial IDEFIX Model
(Illustration only)

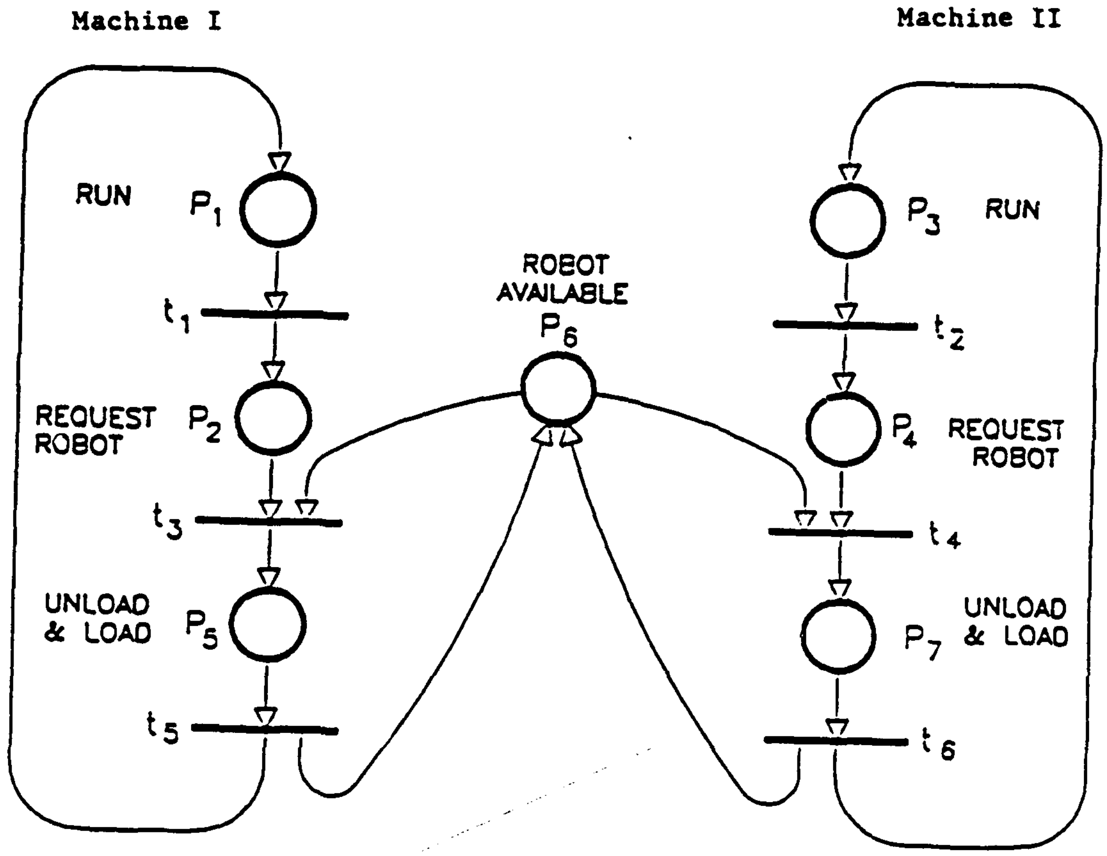


Figure 6 -- A Petri net for a manufacturing cell with two machines and a robot for material handling. The marked places specify the current state of the cell.

COMBAT RATION ADVANCED MANUFACTURING TECHNOLOGY DEMONSTRATION (CRAMTD)

Technical Report: Functional Architecture for Packaged Food Manufacturing

Technical Working Paper (TWP) 37

T.O. Boucher, M.A. Jafari, S. Kim, and J. McPhail
Department of Industrial Engineering
Rutgers University

December 1991

Sponsored by:
DEFENSE LOGISTICS AGENCY
Cameron Station
Alexandria, VA 22304-6145

Contractor:
Rutgers, The State University of New Jersey
THE CENTER FOR ADVANCED FOOD TECHNOLOGY*
Cook College
N.J. Agricultural Experiment Station
New Brunswick, New Jersey 08903

DR. JACK L. ROSSEN
Program Director, CRAMTD
DR. JOHN F. COBURN
Associate Director, CRAMTD

TEL: 908-932-7985
FAX: 908-932-8690

*A New Jersey Commission on Science and Technology Center

"Functional Architecture for
Packaged Food Manufacturing"

Technical Working Paper (TWP) 37

T.O. Boucher, M.A. Jafari, S. Kim and J. McPhail

Abstract

This report describes the Functional Architecture to be used for designing the data base of the CRAMTD Computer Integrated Manufacturing (CIM) demonstration facility. It is a generic architecture and can be used by any company in the packaged food industry that wishes to undertake a CIM implementation project. Particular attention was given to the operating practices of combat ration manufacturers, however, the authors also specify within the architecture practices of civilian product manufacture

Since the architecture is generic, the model is written at a fairly high level of abstraction. In order to give some detail to the reader on how to use the generic architecture for a product specific case, two case studies are included as appendices: MRE pouch - Omelet with Ham, Tray pack - Beef Chunks and Gravy. Only abridged appendices, however, are included in the Technical Working Paper. The complete report of 696 pages is available on special request.

1.0 INTRODUCTION

This report addresses the requirements of Task Item section 3.5.4 of STP #4, requiring a technical report on the design of a functional architecture for packaged food manufacturing. Phase II of STP #4 required studying the procedures by which coalition companies operated their enterprises in the manufacture of shelf stable food products. From these studies the research team was required to abstract the common features of the coalition companies studied in order to describe a generic set of operating procedures. This generic model is referred to as a "Functional Architecture". A Functional Architecture is a description of the functions performed in operating the enterprise and the relations between these functions as given by the information flows and material flows linking them. From these studies it was further required to describe the Functional Architecture as it will be implemented in a Computer Integrated Manufacturing (CIM) environment. This meant paying particular attention to information flows and how they would be captured in a factory data base. It also meant redesigning or respecifying some functions to take advantage of the data base environment that would exist in a CIM plant. The architecture of the observed (existing) manufacturing plants is called the "AS IS" architecture. The architecture that results from redesigning functions and procedures is called the "TO BE" architecture. In this report we present the "TO BE" architecture; that is, the architecture to be used for designing the data base of the CRAMTD demonstration facility. It is a generic architecture and can be

used by any company in the package food industry that wishes to undertake a CIM implementation project.

Particular attention was given to the operating practices of combat ration manufacturers. However, we also specify within the architecture the practices of civilian product manufacture. The overlap is quite considerable.

Since the architecture is generic, the model is written at a fairly high level of abstraction. For most functions it should not be difficult for the reader to relate the model to his or her own enterprise. However, at the factory floor level, a generic description can be difficult to interpret because manufacturing steps are very product specific and a "generic" architecture tries not to be product specific. In order to give some detail to the reader on how to use the generic architecture for a product specific case, we have prepared two appendices for this report. Each appendix describes a specific example of the shop floor level architecture. The hypothetical case study of appendix I is that of a MRE pouch manufacturer producing Omelet with Ham. The hypothetical case study of appendix II is that of a Tray Pack manufacturer producing Beef Chunks and Gravy. The first case study is based on practices currently used in some contractor plants observed by the research team during this study. The case study is meant to be illustrative of how the Functional Architecture design methodogy would be employed in studying current practices. The second case study was carried out in the CRAMTD pilot plant and applied the methodology to the automated environment of that production system as it currently

exists.

The text of this document contains diagrams and descriptions using the IDEFO methodology. IDEFO (Integrated Computer-Aided Manufacturing Definition O) is a modeling procedure developed under funding of the U.S. Air Force. It is an extension of an earlier modeling technique called SADT (Structured Analysis and Design Technique) developed by SoftTech, Inc. of Waltham, Massachusetts.

The IDEFO models were documented on computer. The software that was used is a product of Knowledge Based systems, Inc. of College Station, Texas. It is called AIO.

In the next section we will give an overview of the IDEFO (SADT) methodology. This will be followed by a description of the format of the AIO documentation.

2.0 SADT AND IDEFO

Systems model building is an established area in computer science and management information system design. In Computer Integrated Manufacturing an often used method is the "Structured Analysis and Design Technique" (SADT) or its descendent, IDEFO (Integrated Computer-Aided Manufacturing Definition O), developed under U.S. Air Force sponsorship by SoftTech, Inc. SADT or IDEFO is a modeling methodology for designing and documenting hierarchic, layered, modular systems.

The building block of this modeling approach is the activity box, shown in Figure 1. The Activity Box defines a specific activity in the organization that is being modeled. The Activity may be a decision making or information conversion activity or a

material conversion activity. Inputs to the activity are shown at the left of the box. Inputs are items (material, information) that are transformed by the activity. Outputs of the activity are shown at the right of the box. Outputs are the results of the activity acting on the inputs. Controls are shown entering the Activity box from the top. A control is a condition that governs the performance of the activity. For example, a control may be a set of rules governing the activity or a condition that must exist before the activity can begin. Mechanisms enter the activity box from the bottom. A mechanism is the means by which an activity is realized. For example, a mechanism may be a machine or a worker.

The activity box and the four entities of Figure 1 provide a concise expression: An <u>input</u> is transformed into an <u>output</u> by an <u>activity</u> performed by a <u>mechanism</u> and governed by a <u>control</u>. The specific activity, its inputs, outputs, mechanisms, and controls must be defined for the situation being modeled. Activity boxes represent actions being performed and are labeled with verb phrases. Inputs, outputs, controls, and mechanisms are things, and are labeled with noun phrases.

SADT is applied using top down hierarchic decomposition. This is illustrated in Figure 2. At the top of the hierarchy is the overall purpose of the model; it is the global activity that is the subject of the model. The overall activity is decomposable into components that, when taken together, comprise the global activity. This is the second tier of the hierarchy. Similarly, the second tier activities may be further decomposed

into component activities. The decomposition process continues until there is sufficient detail to serve the purpose of the model builder.

Models are coordinated sets of diagrams. Each layer of the model is coordinated with its sublayers through inputs, outputs, controls, and mechanisms. An example is shown in figures 3 and 4. figure 4 is a sublayer of figure 3. Note that the inputs, outputs, controls, and mechanisms that are at the boundary of figure 3 are also at the boundary of figure 4. In this manner the diagrams are made consistent and material flows and information flows are trackable throughout the model. In some instances it is more convenient to specify mechanisms only at the lowest sublayer. We have followed that practice in our model development.

This brief description will assist the reader in understanding this document. For more details, the reader is referred to the following publication: Structured analysis and Design Technique by D.A. Marca and C.L. McGowan, McGraw-Hill Book Company, 1988.

3.0 SOFTWARE DOCUMENTATION

The documentation of the IDEFO model begins with the highest level activity: "Operate a Shelf Stable Food Manufacturing Enterprise". At each stage of the layered architecture an activity is defined, followed by a breakdown diagram of the subactivities that comprise the major activity. This is followed by a "glossary". The glossary contains the definition of all the "concepts" used in the diagram. "Concepts" are the names

attached to input, output, control and mechanism arrows. Finally, this is followed by a definition of each of the activity blocks of the breakdown diagram of subactivities. By methodically proceeding through the text, the reader can review diagrams, read concept definitions, and read activity definitions.

The next section is the text of the generic architecture. This is followed by Appendix I and Appendix II, which are two case studies showing examples of shop floor diagrams for specific products.

Due to the size of the generic architecture model and software limitations, it was necessary to divide it in half for documentation. The generic architecture includes four major enterprise activities:

1) Manage Contracts, Orders, and Bidding Process
2) Plan for Manufacture
3) Manufacture Product
4) Control Manufactured Product

The first two major enterprise activities are documented in Section A. Major activities 3 and 4 are documented in Section B.

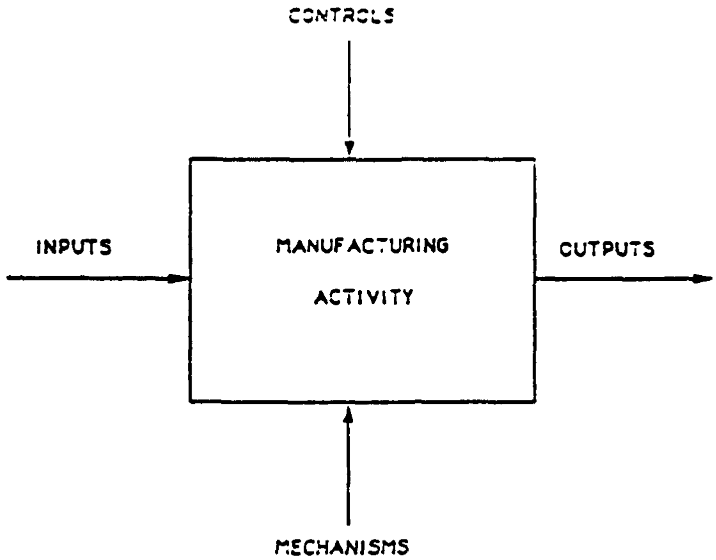


Figure 1. SADT ACTIVITY BOX AND CONNECTING ARROWS

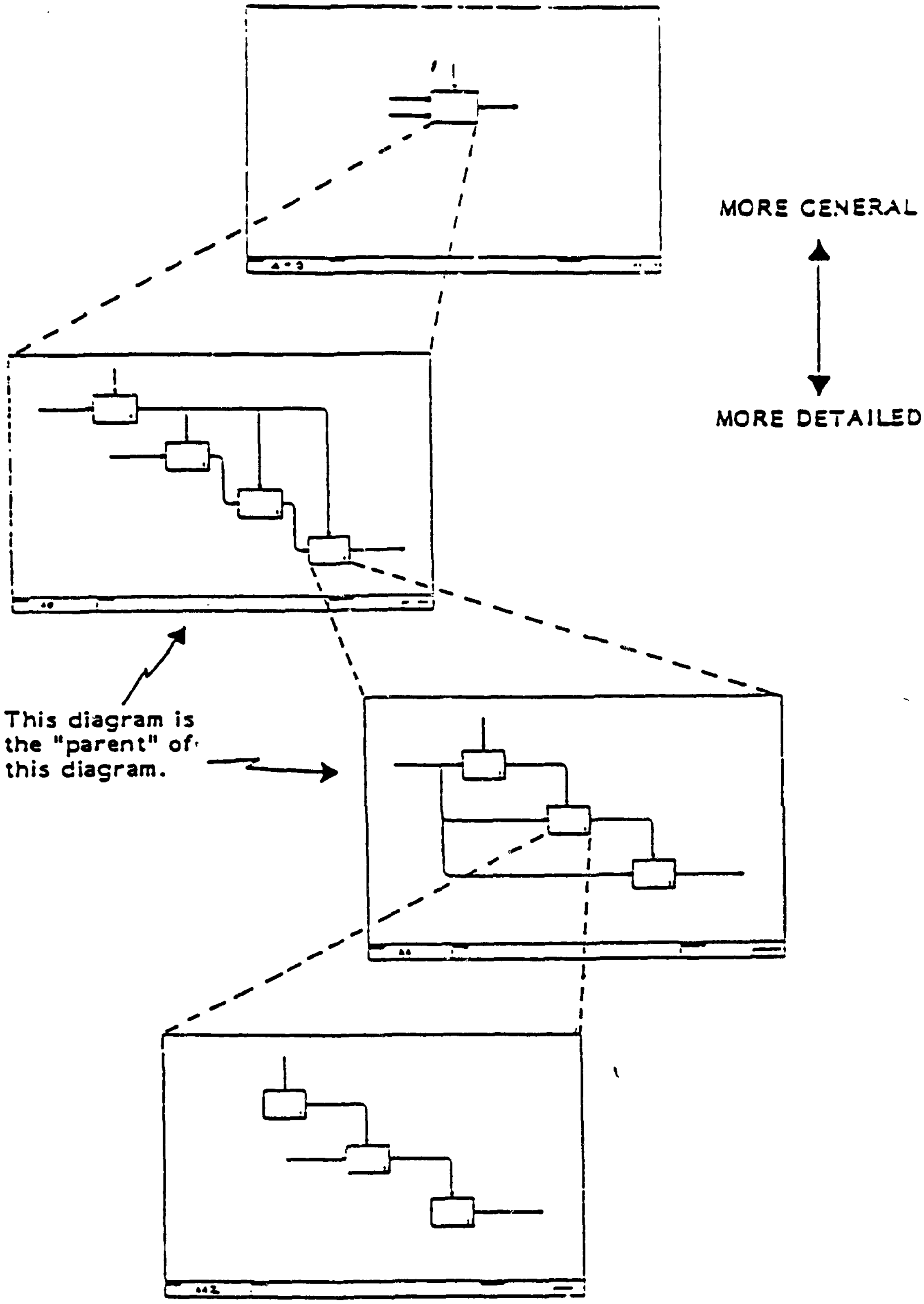


Figure 2.-- Relationship between levels in IDEF0 Methodology

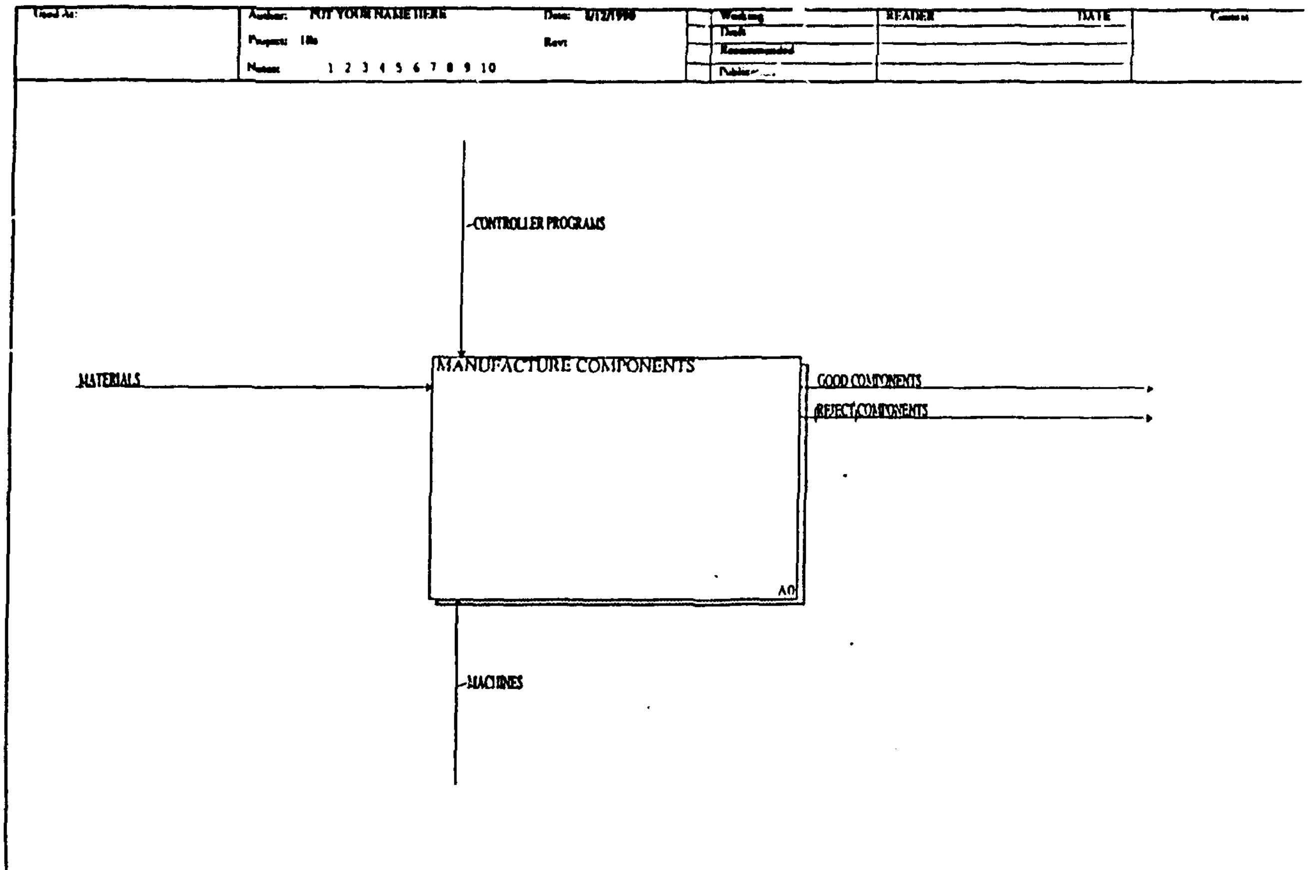


Figure 3. Overall concept diagram

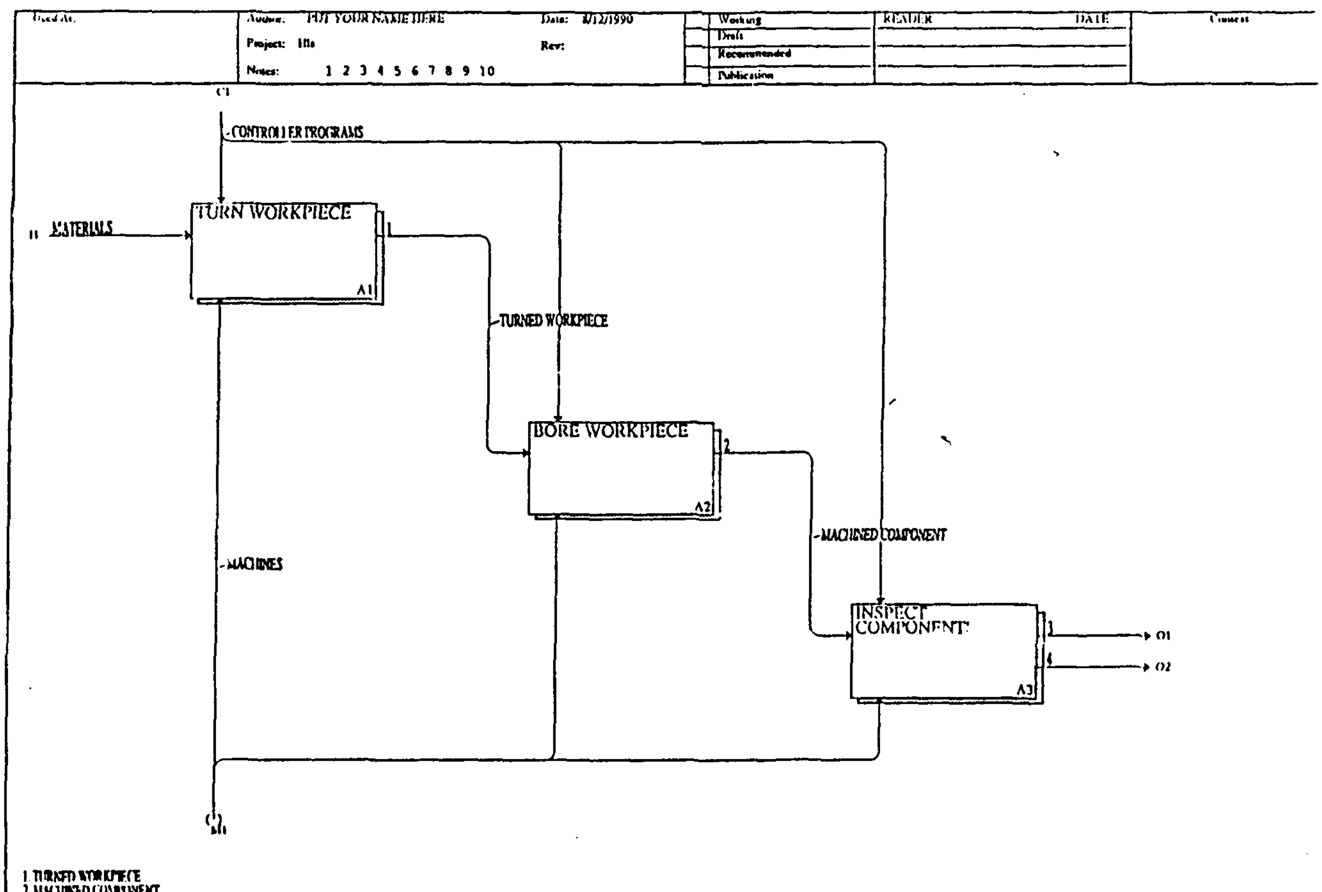


Figure 4. Manufacture components

SECTION A

GENERIC ARCHITECTURE

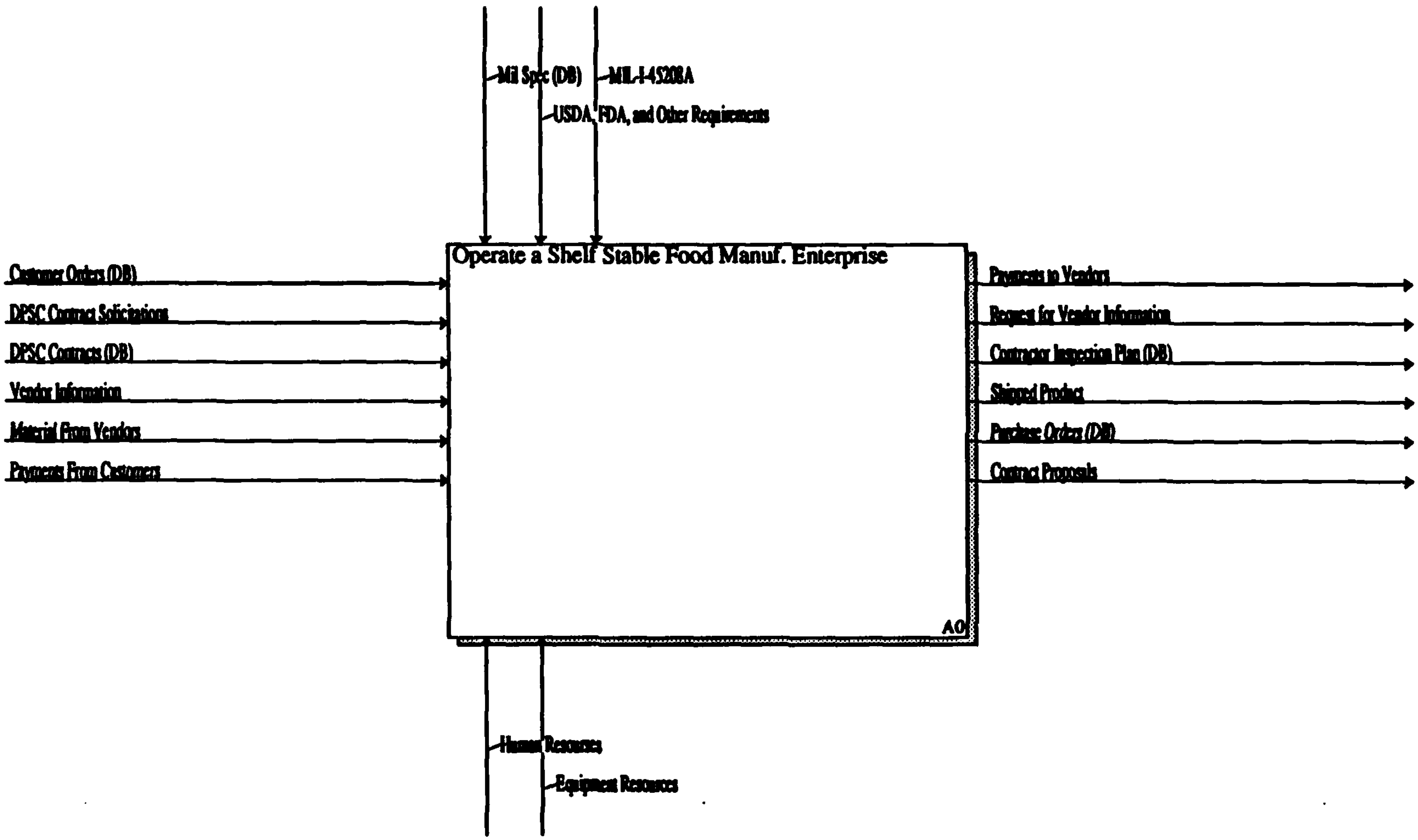
ENTERPRISE ACTIVITIES
(ABRIDGED)

1. Manage Contracts, Orders, and Bidding Process (A1)
2. Plan for Manufacture (A2)

TABLE OF CONTENTS

DIAGRAMS

Used At:	Author: PUT YOUR NAME HERE	Date: 1/2/1992	Working	READER	DATE	Context
	Project: Manuf_en	Rev:	Draft			
	Notes: 1 2 3 4 5 6 7 8 9 10		Recommended			
			Publication			



Node:A-0 | Title:Operate a Shelf Stable Food Manuf. Enterprise (Context) | Number: 17

GLOSSARY

Attached Concepts

Contract Proposals

A contractor's response to a government solicitation to bid on the manufacture of combat rations. It includes quantities to be delivered by time period, bid price, and relevant planning documents as required by the contract solicitation.

Contractor Inspection Plan (DB)

A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and final product inspection.

Customer Orders (DB)

Orders for the production and sale of civilian food product to customers.

DPSC Contract Solicitations

Requests to bid on the production and sale of combat rations. Includes identification of product, total units required, proposed delivery schedule, and other requirements to be met by the contractors.

DPSC Contracts (DB)

The final award to a contractor indicating the quantities of each product that the government will purchase.

Equipment Resources

Machines and other equipment employed by the manufacturing enterprise in operating its business.

Human Resourses

Employees of the manufacturing enterprise.

Material From Vendors

Primary raw materials, services, equipment, and supplies converted by or consumed in the manufacturing process.

Mil Spec (DB)
The specification for manufacturing the combat ration. Includes ingredients by weight, preparation procedures, quality assurance provisions, and packaging requirements.

MIL-I-45208A
Military specification for preparation of Contractor Inspection Plan.

Payments From Customers
Payments received for the delivery of shipped product.

Payments to Vendors
Payments made for the receipt of materials and services from vendors.

Purchase Orders (DB)
A document that initiates the sale of material from a vendor to the manufacturing enterprise.

Request for Vendor Information
Inquiries from manufacturing enterprise to vendor; e.g., requests for current pricing and delivery lead times of materials.

Shipped Product
Rations or civilian products produced to specification and shipped to appointed destination.

USDA, FDA, and Other Requirements
Includes requirements imposed on the manufacture of food products by federal and state agencies, for example, the USDA meat and poultry inspection regulations.

Vendor Information
Responses to requests for vendor information.

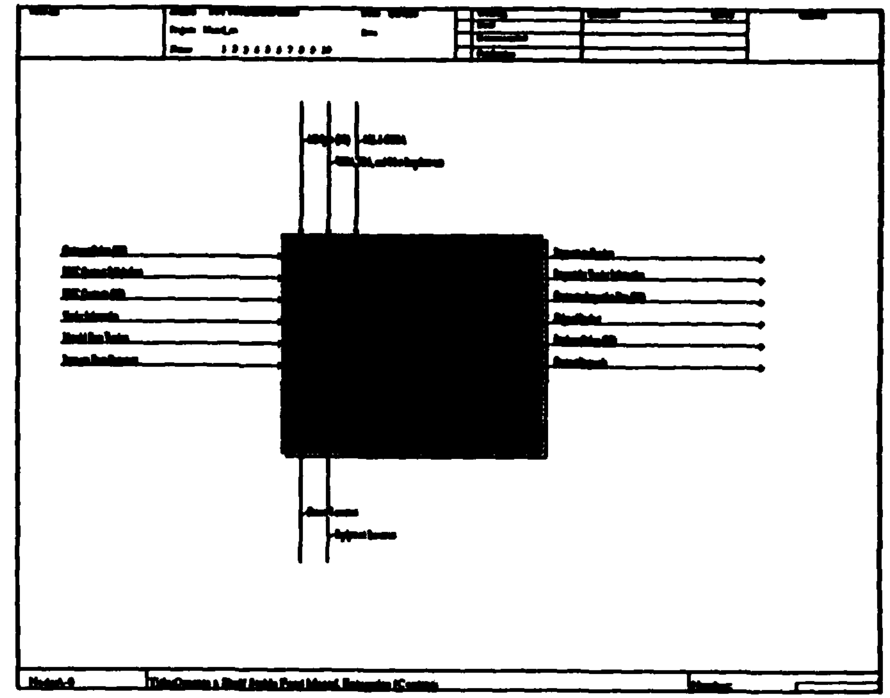
Operate a Shelf Stable Food Manuf. Enterprise

A0 is the root function diagram that describes the activities involved to operate a Shelf Stable Food Manufacturing Enterprise. The inputs, outputs, and controls at this level are interfaces between the enterprise and the outside world. The Shelf Stable Food Manufacturing Enterprise includes manufacture of MRE Pouch entres, manufacture of Tray Pack entres, and manufacture of shelf stable packaged civilian products.

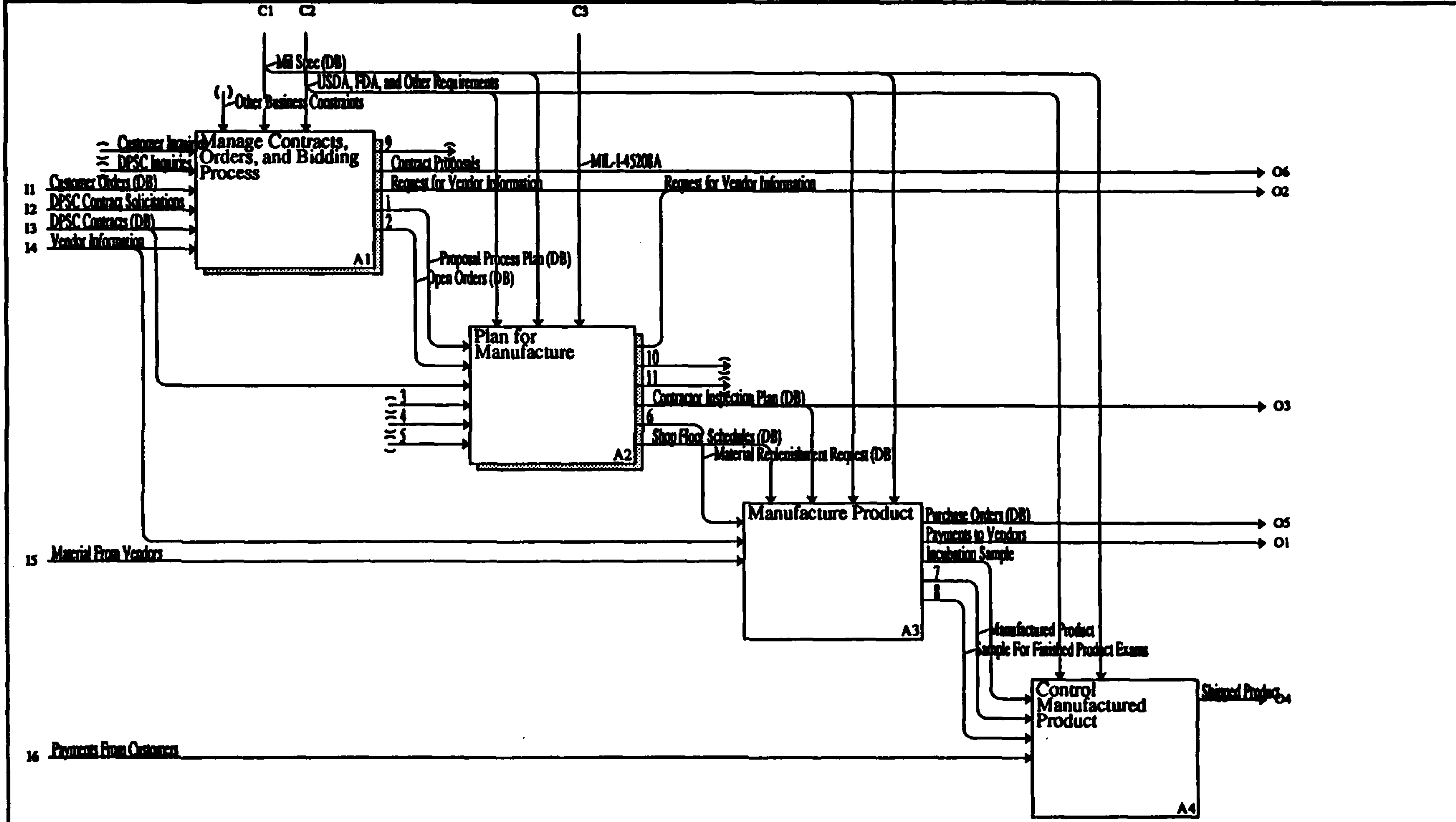
There are two pre-production activities. The "Manage Contracts, Orders, and Bidding Process" is the sales and contracting interface with customers. For civilian customers, it takes "Customer Inquirires" as input and provides "Response to Customers" as outputs. For DSPC, it takes "Contract Solicitations" as input and provides "Contract Proposals" as output. This function also performs order entry for final orders and contract awards.

The "Plan for Manufacture" function begins with the awarding of a contract or the issuing of an order/forecast. This function includes developing a Manufacturing Plan and Production Schedule and the approval of a Contractor Inspection Plan.

There are two production activities. "Manufacture Product" includes the receipt of ingredients and other materials, ingredient and material inspection, ingredient processing and packaging, and in-process control of product and process. "Control Manufactured Product" includes finished product inspections, rework and re-inspection, storage, handling and shipping.



Used At:	Author: PUT YOUR NAME HERE	Date: 1/2/1992	Working / Draft / Recommended / Publication
	Project: Manuf_en	Rev:	READER / DATE
	Notes: 1 2 3 4 5 6 7 8 9 10		Context:



1. Proposal Process Plan (DB)
2. Open Orders (DB)
3. Vendor Lot Inventory (DB)
4. Finished Goods Inv. (Labeled & Unlabled)(DB)
5. New Product Orders
6. Material Replenishment Request (DB)
7. Manufactured Product
8. Sample For Finished Product Exams
9. Response to Customer
10. Contract Aggregate Production Plan (DB)
11. Month Production Schedule (DB)

Node: A0	Title: Operate a Shelf Stable Food Manuf. Enterprise	Number: 21

GLOSSARY

Attached Concepts

Contract Aggregate Production Plan (DB)

A plan for production, manpower, and material requirements by time period that, when executed, should enable the enterprise to meet its contract shipping schedule. It consists of a manufacturing schedule, a materials plan, a personnel plan, and a distribution plan associated with the production of a specific contract for combat rations.

Contract Proposals

A contractor's response to a government solicitation to bid on the manufacture of combat rations. It includes quantities to be delivered by time period, bid price, and relevant planning documents as required by the contract solicitation.

Contractor Inspection Plan (DB)

A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and final product inspection.

Customer Inquiries

Requests from customers for pricing and/or delivery lead times on products, status of current orders, or status of evaluation of new product proposals.

Customer Orders (DB)

Orders for the production and sale of civilian food product to customers.

DPSC Contract Solicitations

Requests to bid on the production and sale of combat rations. Includes identification of product, total units required, proposed delivery schedule, and other requirements to be met by the contractors.

DPSC Contracts (DB)

The final award to a contractor indicating the quantities of each product that the government will purchase.

DPSC Inquiries

Requests from the contracting agency concerning the status of contract execution.

Equipment Resources
Machines and other equipment employed by the manufacturing enterprise in operating its business.

Finished Goods Inv. (Labeled & Unlabled)(DB)
Manufactured product that is in compliance with quality specifications and is in storage. May be stored with a customer Label or in the unlabled (shiner) condition.

Human Resourses
Employees of the manufacturing enterprise.

Incubation Sample
A sample of product drawn from each retort basket after retorting. Held for ten days and examined for container swelling. Part of the process of insuring that the package contents are properly sterilized.

Manufactured Product
Rations or civilian product that have completed the production processes and are awaiting clearence for acceptable quality.

Material From Vendors
Primary raw materials, services, equipment, and supplies converted by or consumed in the manufacturing process.

Material Replenishment Request (DB)
A request to the materials manager for the procurement of material from a supplier to the manufacturing enterprise.

Mil Spec (DB)
The specification for manufacturing the combat ration. Includes ingredients by weight, preparation procedures, quality assurance provisions, and packaging

requirements.

MIL-I-45208A
Military specification for preparation of Contractor Inspection Plan.

Month Production Schedule (DB)
A medium range aggregate production schedule which specifies the products to be produced for the next four weeks. Revised weekly.

New Product Orders
Requests for the manufacture of a new civilian product.

Open Orders (DB)
The existing backlog of unfilled orders and internal orders (forecasts).

Other Business Constraints
Factors that constrain a particular enterprise in its participation in combat ration contracts or other new business. May include available capital, level of past experience with a particular type of product, and/or other factors of a general business nature.

Payments From Customers
Payments received for the delivery of shipped product.

Payments to Vendors
Payments made for the receipt of materials and services from vendors.

Proposal Process Plan (DB)
The manufacturer's identification of the specific processes that will be used in the manufacture of the ration product. This plan, though not submitted as part of the proposal, is one of the bases for estimating manufacturing costs and production quantities.

Purchase Orders (DB)

A document that initiates the sale of material from a vendor to the manufacturing enterprise.

Request for Vendor Information

Inquiries from manufacturing enterprise to vendor; e.g., requests for current pricing and delivery lead times of materials.

Response to Customer

Response to a customer inquiry as defined in attached concept "Customer Inquiry".

Sample For Finished Product Exams

A sample of product taken from production and used for examining the quality of the finished product. Includes a container evaluation and a product evaluation.

Shipped Product

Rations or civilian products produced to specification and shipped to appointed destination.

Shop Floor Schedules (DB)

A group of schedules that are provided daily to the line supervisor for executing production. They include the "Day Production Schedule", "Material Move Schedule and Report", "Raw Preparation Sheet", "Cook Sheet", and "Daily Process Information".

USDA, FDA, and Other Requirements

Includes requirements imposed on the manufacture of food products by federal and state agencies, for example, the USDA meat and poultry inspection regulations.

Vendor Information

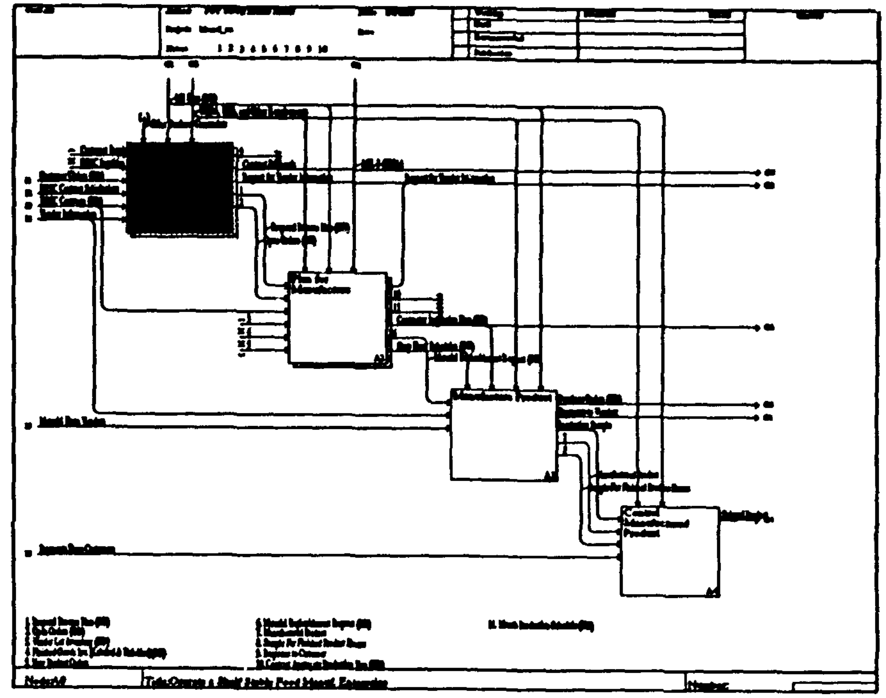
Responses to requests for vendor information.

Vendor Lot Inventory (DB)

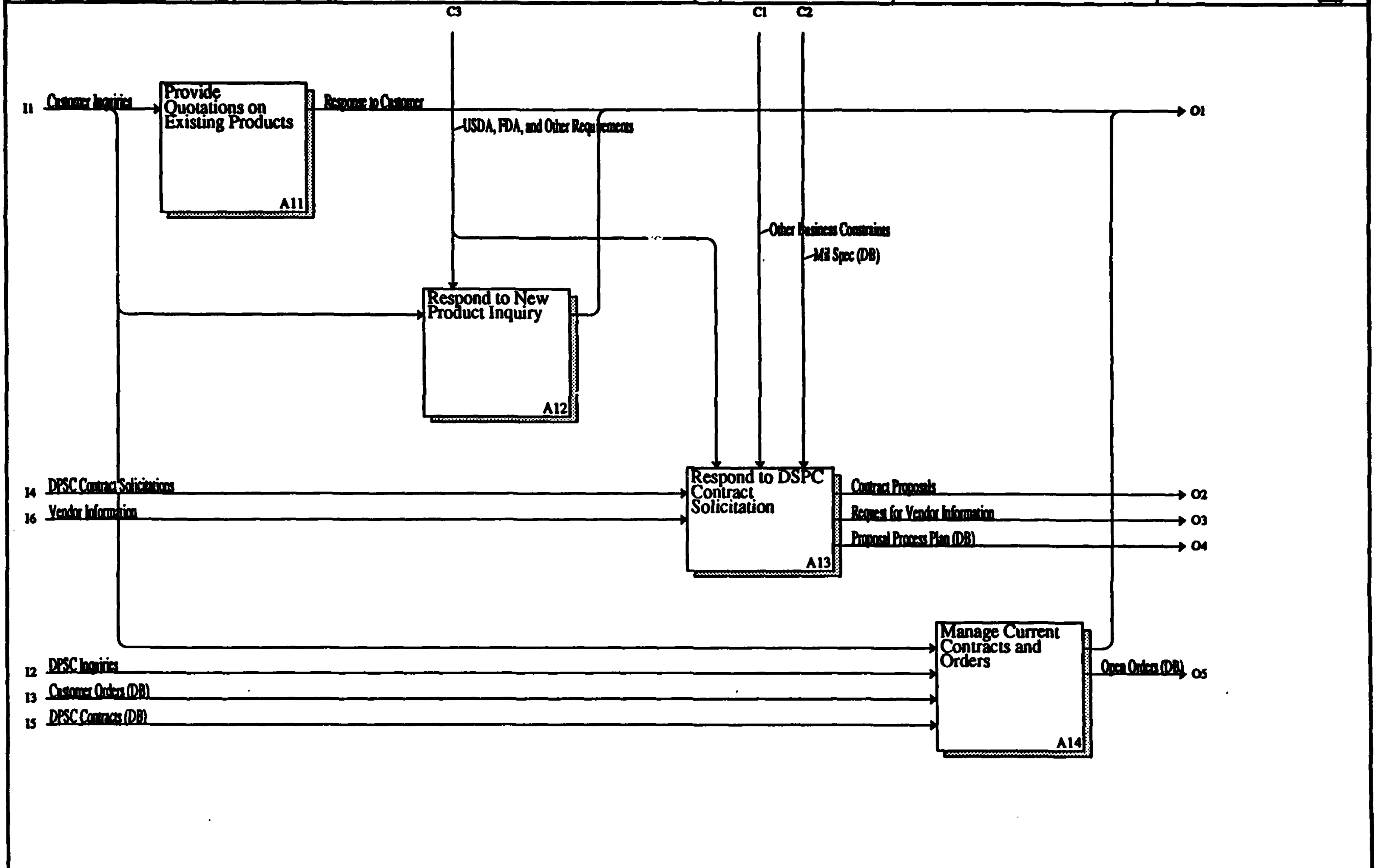
The inventory of available raw materials, which are recorded and identifiable by vendor lot numbers.

Manage Contracts, Orders, and Bidding Process

"Manage Contracts, Orders, and Bidding Process" includes four functions: "Provide Quotations on Existing Products", "Respond to New Product Inquiry", Respond to DSPC Contract Solicitation" ,and "Manage Current Contracts and Orders". These functions represent the interface with the customer on the selling side and would normally be the responsibility of sales. In government or other long term contracting, the activities are often handled by an administrator charged with overseeing the contract. Node A14 ("Manage Current Contracts and Orders") provides a tracking mechanism for monitoring the status of all open orders and active contracts.



Used At:	Author: PUT YOUR NAME HERE	Date: 1/2/1992	Working	READER	DATE	Context
	Project: Manuf_m	Rev:	Draft			
	Notes: 1 2 3 4 5 6 7 8 9 10		Recommended			
			Publication			



Node:A1	Title:Manage Contracts, Orders, and Bidding Process	Number: 27

GLOSSARY

Attached Concepts

Contract Proposals

A contractor's response to a government solicitation to bid on the manufacture of combat rations. It includes quantities to be delivered by time period, bid price, and relevant planning documents as required by the contract solicitation.

Customer Inquiries

Requests from customers for pricing and/or delivery lead times on products, status of current orders, or status of evaluation of new product proposals.

Customer Orders (DB)

Orders for the production and sale of civilian food product to customers.

DPSC Contract Solicitations

Requests to bid on the production and sale of combat rations. Includes identification of product, total units required, proposed delivery schedule, and other requirements to be met by the contractors.

DPSC Contracts (DB)

The final award to a contractor indicating the quantities of each product that the government will purchase.

DPSC Inquiries

Requests from the contracting agency concerning the status of contract execution.

Mil Spec (DB)

The specification for manufacturing the combat ration. Includes ingredients by weight, preparation procedures, quality assurance provisions, and packaging requirements.

Open Orders (DB)

The existing backlog of unfilled orders and internal orders (forecasts).

Other Business Constraints

Factors that constrain a particular enterprise in its participation in combat ration contracts or other new business. May include available capital, level of past experience with a particular type of product, and/or other factors of a general business nature.

Proposal Process Plan (DB)

The manufacturer's identification of the specific processes that will be used in the manufacture of the ration product. This plan, though not submitted as part of the proposal, is one of the bases for estimating manufacturing costs and production quantities.

Request for Vendor Information

Inquiries from manufacturing enterprise to vendor; e.g., requests for current pricing and delivery lead times of materials.

Response to Customer

Response to a customer inquiry as defined in attached concept "Customer Inquiry".

USDA, FDA, and Other Requirements

Includes requirements imposed on the manufacture of food products by federal and state agencies, for example, the USDA meat and poultry inspection regulations.

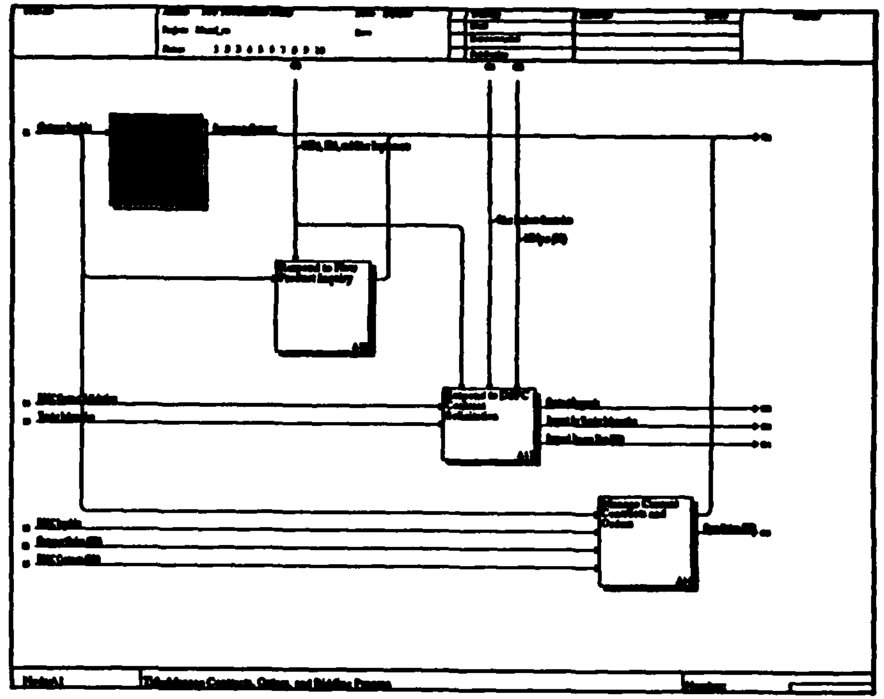
Vendor Information

Responses to requests for vendor information.

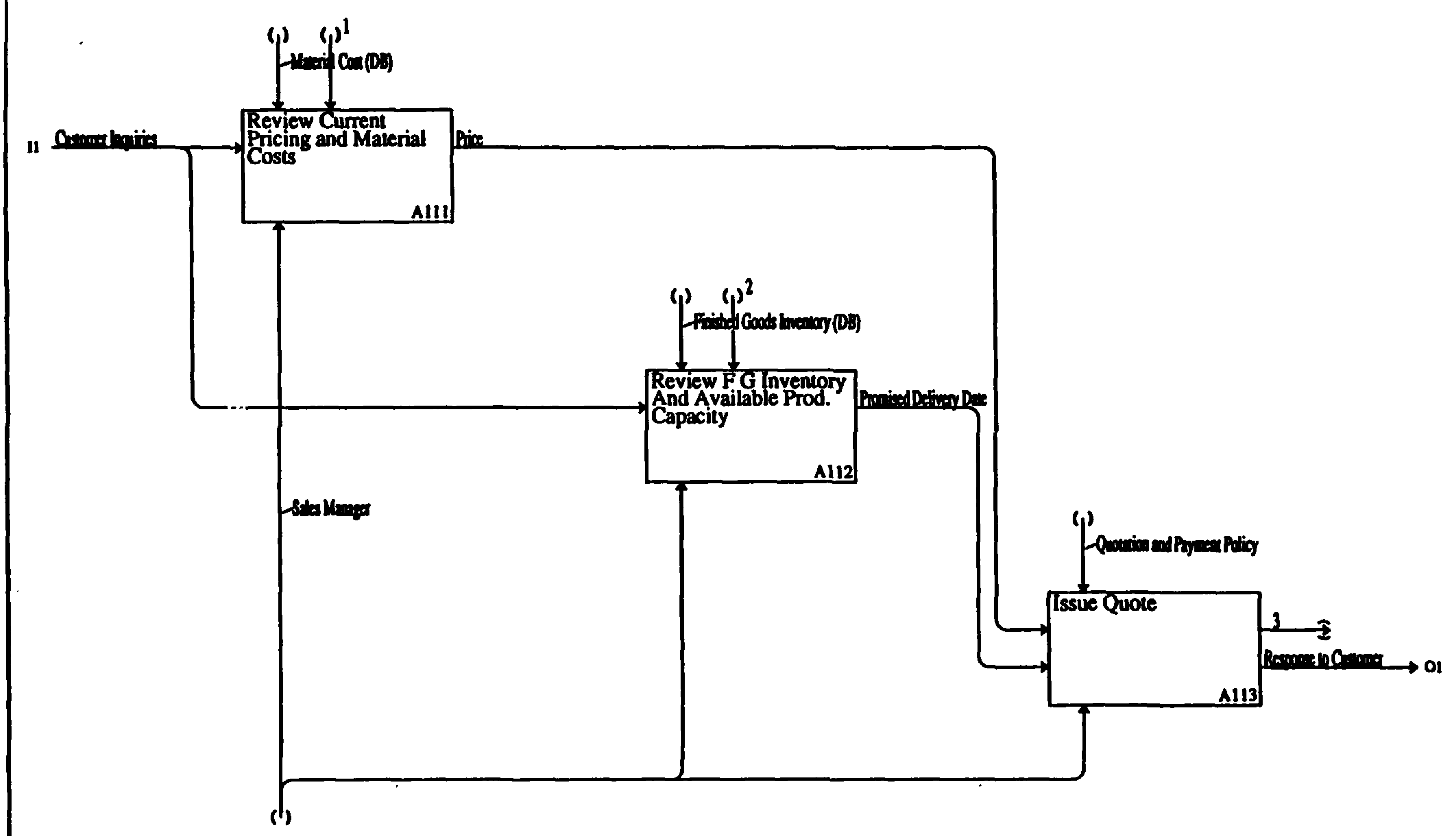
Provide Quotations on Existing Products

This activity is a response to a customer inquiry concerning pricing and delivery time on existing product. This is part of the firm's sales activities. It begins with referring to the current published price list to establish price ("Review Current Pricing and Material Cost"). The delivery lead time is determined by reviewing "Finished Goods Inventory and Available Production Capacity". The "Issue Quote" function combines price and lead time with other policy information and provides a written or a verbal "Response to Customer". A record is kept ("Quotation(DB)").

The breakdown of Node A11 is the first time in this text that the reader encounters activities at the bottom of the hierarchy. When such activities are encountered, the authors have indicated data used in the execution of the activity that comes from the preliminary CIM database by the suffix "(DB)". The glossary of attached concepts will provide more detail.



Used At:	Author: PUT YOUR NAME HERE	Date: 1/2/1992	Working	READER	DATE	Context
	Project: Manuf_en	Rev:	Draft			
	Notes: 1 2 3 4 5 6 7 8 9 10		Recommended			
			Publication			



1. Price List (DB)
2. Available Production Capacity (DB)
3. Quotation (DB)

Node: A11	Title: Provide Quotations on Existing Products	Number: 31

GLOSSARY

Attached Concepts

Available Production Capacity (DB)
Uncommitted time on a specific production (filling) line. It is determined from the current order backlog, order delivery dates, and total manufacturing capacity.

Customer Inquiries
Requests from customers for pricing and/or delivery lead times on products, status of current orders, or status of evaluation of new product proposals.

Finished Goods Inventory (DB)
Current record of labeled and unlabled finished product.

Material Cost (DB)
Record of standard cost and actual last price paid.

Price
Quotation price of product based on units ordered.

Price List (DB)
Current prices established by the company for specific products ordered in specific quantities.

Promised Delivery Date
Quotation delivery date for proposed units to be ordered.

Quotation (DB)
An electronic record of a quotation issued to a customer.

Quotation and Payment Policy
Company policies with respect to such items as the length of time for which a quotation is effective or the accounts receivable payment and discount policy.

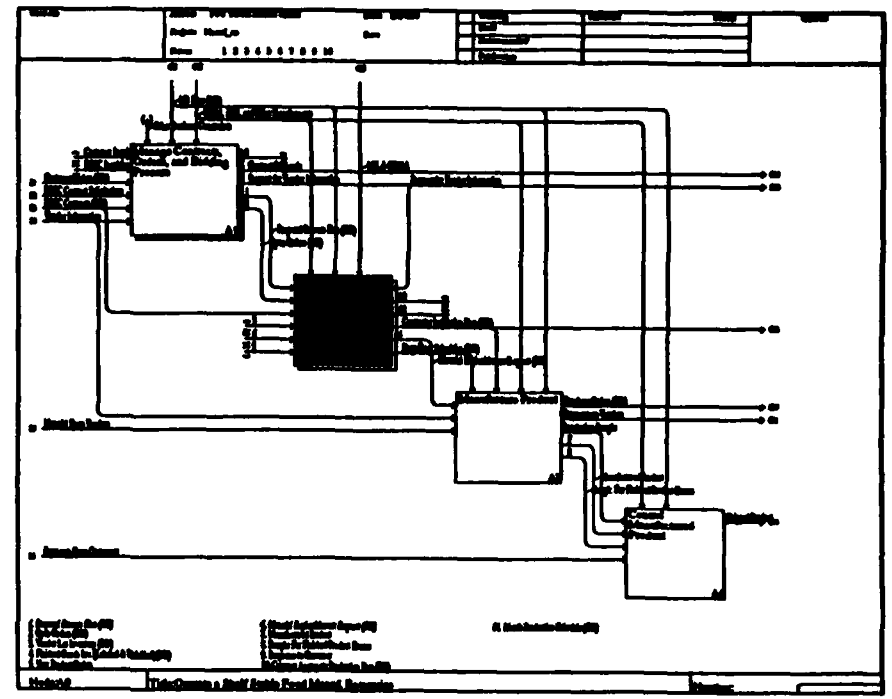
Response to Customer

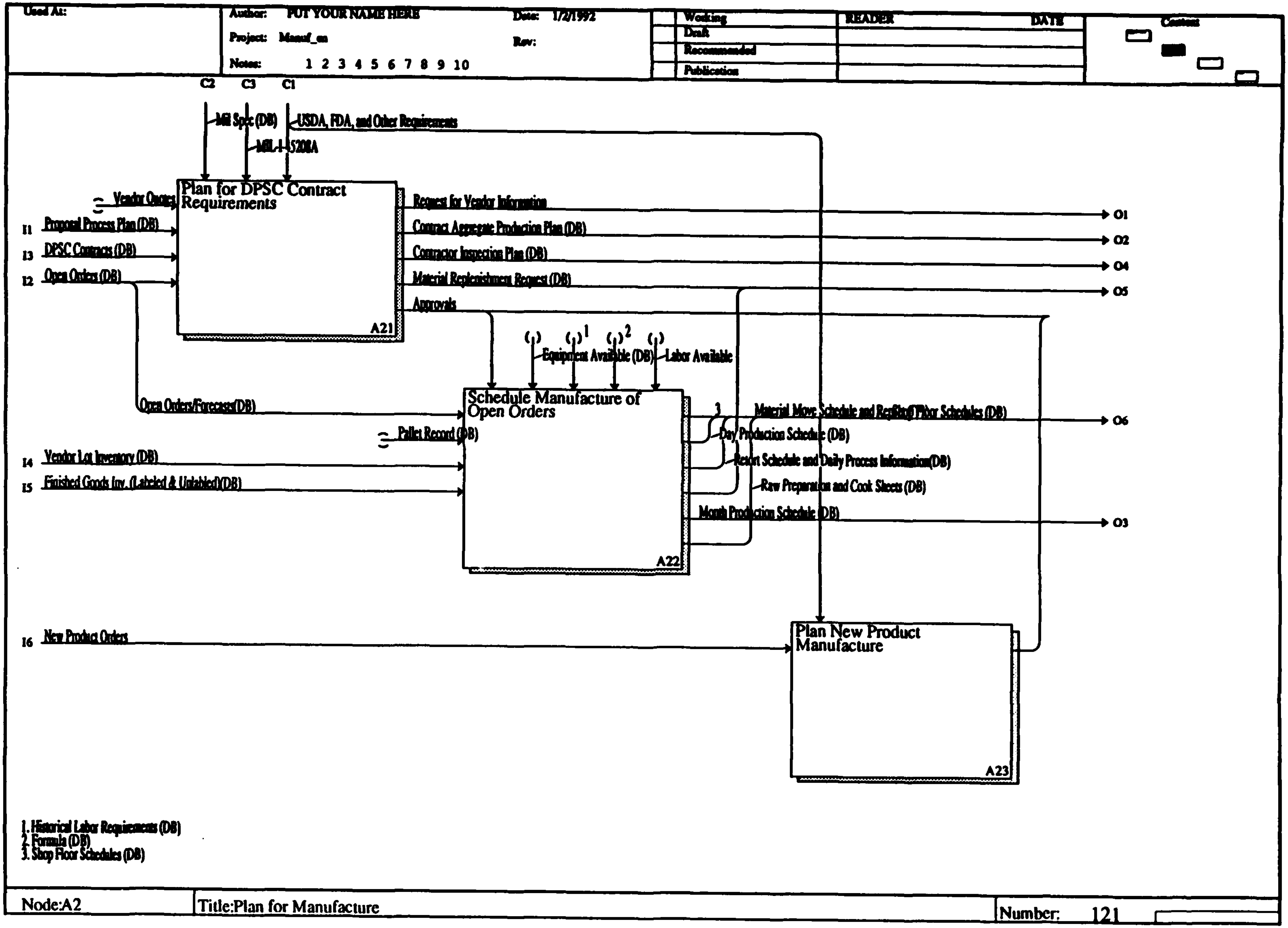
Response to a customer inquiry as defined in attached concept "Customer Inquiry".

Sales Manager
The individual who handles customer relationships.

Plan for Manufacture

"Plan for Manufacture" represents the second of the four major activities in Operating a Shelf Stable Food Manufacturing Enterprise. "Plan for Manufacture" begins with the awarding of a combat rations contract or the entry of civilian product orders into the backlog. There are three major sub-activities. "*Plan for DSPC Contract Requirements*" includes all the pre-production activities before a contractor obtains final approval to produce. "Schedule Manufacture of Open Orders" includes the scheduling activities associated with open orders that have cleared all approval processes. Finally, "Plan for New Product Manufacture" includes all activities before a new product can be scheduled into production.




Used At:
Author: PUT YOUR NAME HERE
Date: 1/2/1992
Project: Manuf_an
Rev:
Notes: 1 2 3 4 5 6 7 8 9 10
Working
Draft
Recommended
Publication
READER
DATE
Context
C2
C3
C1
Mil Spec (DB)
MIL-I-45208A
USDA, FDA, and Other Requirements
Plan for DPSC Contract Requirements
A21
Vendor Quotes
I1 Proposal Process Plan (DB)
I3 DPSC Contracts (DB)
I2 Open Orders (DB)
Request for Vendor Information
Contract Aggregate Production Plan (DB)
Contractor Inspection Plan (DB)
Material Replenishment Request (DB)
Approvals
O1
O2
O4
O5
Equipment Available (DB)
Labor Available
Open Orders/Forecasts(DB)
Pallet Record (DB)
I4 Vendor Lot Inventory (DB)
I5 Finished Goods Inv. (Labeled & Unlabled)(DB)
Schedule Manufacture of Open Orders
A22
Material Move Schedule and Replen. Floor Schedules (DB)
Day Production Schedule (DB)
Retort Schedule and Daily Process Information(DB)
Raw Preparation and Cook Sheets (DB)
Month Production Schedule (DB)
O6
O3
I6 New Product Orders
Plan New Product Manufacture
A23
1. Historical Labor Requirements (DB)
2. Formula (DB)
3. Shop Floor Schedules (DB)
Node:A2
Title:Plan for Manufacture
Number: 121

GLOSSARY

Attached Concepts

Approvals

All necessary approvals prior to placing a product into production. May include approved contractor inspection plan, product labeling, first article approval, and so forth.

Contract Aggregate Production Plan (DB)

A plan for production, manpower, and material requirements by time period that, when executed, should enable the enterprise to meet its contract shipping schedule. It consists of a manufacturing schedule, a materials plan, a personnel plan, and a distribution plan associated with the production of a specific contract for combat rations.

Contractor Inspection Plan (DB)

A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and final product inspection.

Day Production Schedule (DB)

The schedule of which products are to be produced on which production equipment for a specific day. This is a firm schedule released the afternoon before the day's production.

DPSC Contracts (DB)

The final award to a contractor indicating the quantities of each product that the government will purchase.

Equipment Available (DB)

A data file of equipment available to be used in production and its characteristics, such as worker requirements, output rate, and so forth.

Finished Goods Inv. (Labeled & Unlabled)(DB)

Manufactured product that is in compliance with quality specifications and is in storage. May be stored with a customer Label or in the unlabled (shiner) condition.

Formula (DB)
A description of the ingredient and packaging content of a product along with processing information.

Historical Labor Requirements (DB)
The number of workers historically required to operate the production line at a specific rate when producing a specific product.

Labor Available
Current workforce size and skill composition.

Material Move Schedule and Report (DB)
A shop floor schedule issued with the "Day Production Schedule" describing what vendor lots (raw material) should be moved from inventory to shop floor locations to accomodate the "Day Production Schedule". Also used to report depletion of raw material.

Material Replenishment Request (DB)
A request to the materials manager for the procurement of material from a supplier to the manufacturing enterprise.

Mil Spec (DB)
The specification for manufacturing the combat ration. Includes ingredients by weight, preparation procedures, quality assurance provisions, and packaging requirements.

MIL-I-45208A
Military specification for preparation of Contractor Inspection Plan.

Month Production Schedule (DB)
A medium range aggregate production schedule which specifies the products to be produced for the next four weeks. Revised weekly.

New Product Orders

Requests for the manufacture of a new civilian product.

Open Orders (DB)
The existing backlog of unfilled orders and internal orders (forecasts).

Open Orders/Forecasts(DB)
The existing backlog of unfilled orders and internal orders (forecasts).

Pallet Record (DB)
A record indicating the contents of a pallet, including product and quantities by retort cook also indicates storage location. Used as a primary record of finished goods inventory.

Proposal Process Plan (DB)
The manufacturer's identification of the specific processes that will be used in the manufacture of the ration product. This plan, though not submitted as part of the proposal, is one of the bases for estimating manufacturing costs and production quantities.

Raw Preparation and Cook Sheets (DB)
Shop floor schedules that list the recipe and cooking instructions for the preparation of raw ingredients and the mixing and cooking of batches.

Request for Vendor Information
Inquiries from manufacturing enterprise to vendor; e.g., requests for current pricing and delivery lead times of materials.

Retort Schedule and Daily Process Information(DB)
Shop floor schedule the describes which retorts will be dedicated to the manufacture of each product on a particular day. Also indicates retort settings and other process information.

Shop Floor Schedules (DB)
A group of schedules that are provided daily to the line supervisor for executing

production. They include the "Day Production Schedule", "Material Move Schedule and Report", "Raw Preparation Sheet", "Cook Sheet", and "Daily Process Information".

USDA, FDA, and Other Requirements

Includes requirements imposed on the manufacture of food products by federal and state agencies, for example, the USDA meat and poultry inspection regulations.

Vendor Lot Inventory (DB)

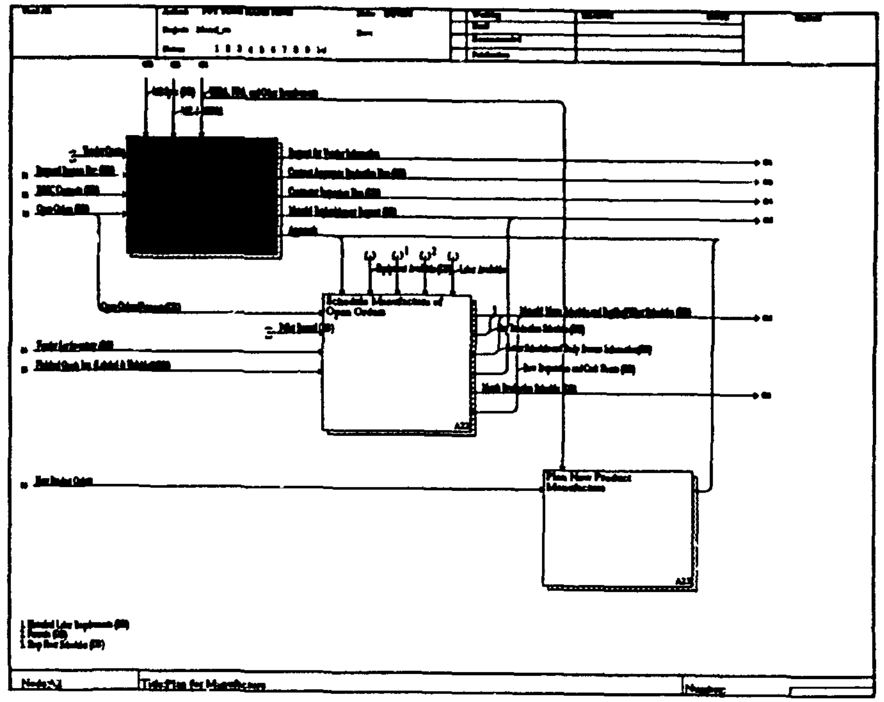
The inventory of available raw materials, which are recorded and identifiable by vendor lot numbers.

Vendor Quotes

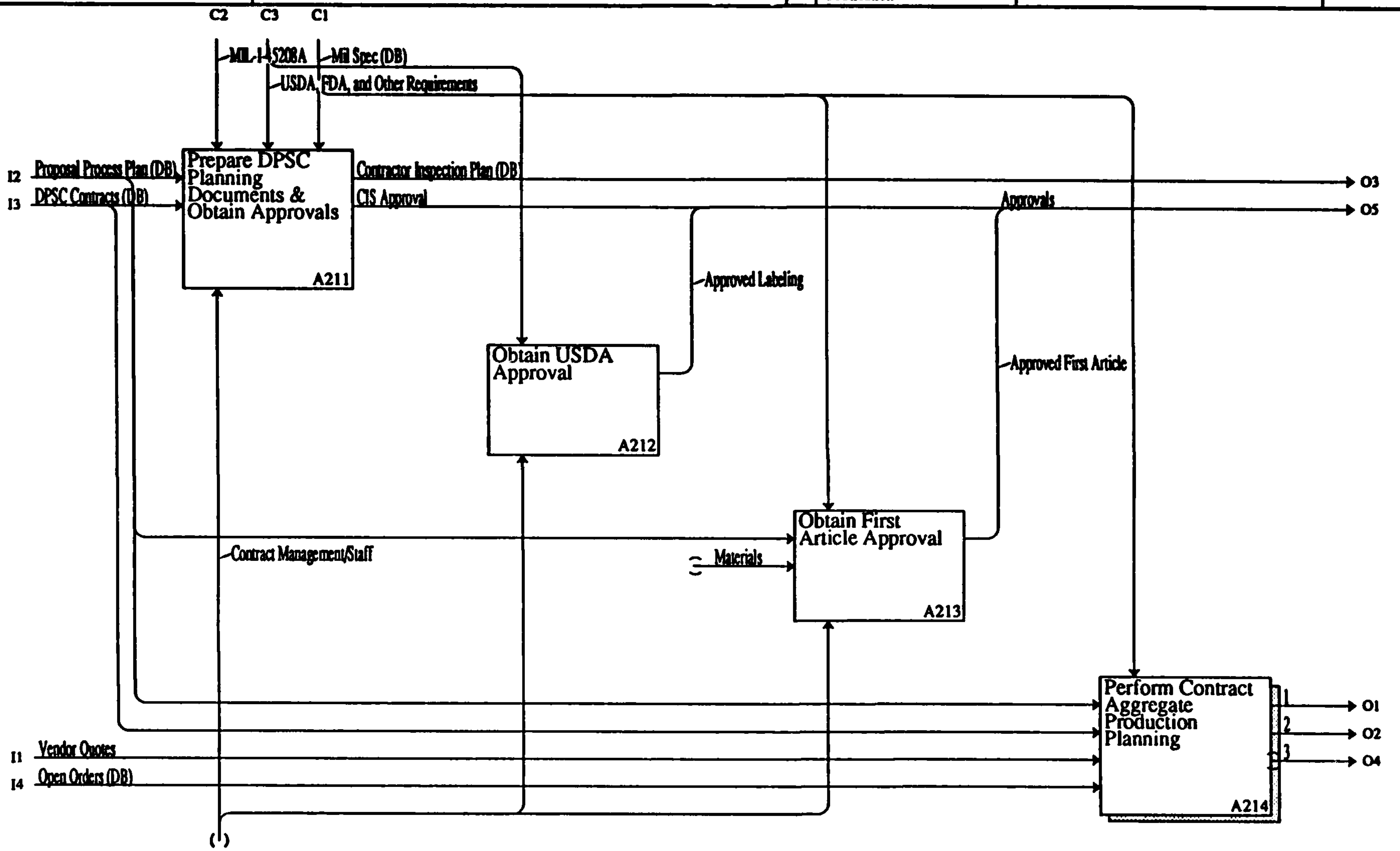
The suppliers' response to "Request for Vendor Information".

Plan for DPSC Contract Requirements

This activity begins after a contract has been awarded and ends prior to the start up of production. The activity includes obtaining approvals required for production and preparing the production facility and a contract aggregate production plan. The specific activities are to "Prepare DPSC Planning Documents and Obtain Approvals", "Obtain USDA Approval", "Obtain First Article Approval", and "Perform Contract Aggregate Production Planning".



Used At:	Author: PUT YOUR NAME HERE Date: 1/2/1992	Working	READER DATE
	Project: Manuf_en Rev:	Draft	
	Notes: 1 2 3 4 5 6 7 8 9 10	Recommended	
		Publication	Context



1. Request for Vendor Information
2. Contract Aggregate Production Plan (DB)
3. Material Replenishment Request (DB)

Node:A21	Title:Plan for DPSC Contract Requirements	Number: 127

GLOSSARY

Attached Concepts

Approved First Article

Notification from the contracting officer of a DPSC contract that the first article submission is of sufficient quality to allow the contractor to begin production of that product against the contract.

Approved Labeling

Product ingredients must be declared on the carton. This ingredient declaration must be approved by the USDA.

CIS Approval

An approval from DPSC informing the contractor that the submitted Contractor Inspection Plan can be used in conjunction with the production of the ration under contract to the enterprise.

Contract Aggregate Production Plan (DB)

A plan for production, manpower, and material requirements by time period that, when executed, should enable the enterprise to meet its contract shipping schedule. It consists of a manufacturing schedule, a materials plan, a personnel plan, and a distribution plan associated with the production of a specific contract for combat rations.

Contract Management/Staff

Individuals with primary responsibility for securing contracts and overseeing the planning of the business.

Contractor Inspection Plan (DB)

A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and final product inspection.

DPSC Contracts (DB)

The final award to a contractor indicating the quantities of each product that the government will purchase.

Material Replenishment Request (DB)

A request to the materials manager for the procurement of material from a supplier to the manufacturing enterprise.

Materials

Packaging and ingredients used in the manufacture of food by the enterpirse.

Mil Spec (DB)

The specification for manufacturing the combat ration. Includes ingredients by weight, preparation procedures, quality assurance provisions, and packaging requirements.

MIL-I-45208A

Military specification for preparation of Contractor Inspection Plan.

Open Orders (DB)

The existing backlog of unfilled orders and internal orders (forecasts).

Proposal Process Plan (DB)

The manufacturer's identification of the specific processes that will be used in the manufacture of the ration product. This plan, though not submitted as part of the proposal, is one of the bases for estimating manufacturing costs and production quantities.

Request for Vendor Information

Inquiries from manufacturing enterprise to vendor; e.g., requests for current pricing and delivery lead times of materials.

USDA, FDA, and Other Requirements

Includes requirements imposed on the manufacture of food products by federal and state agencies, for example, the USDA meat and poultry inspection regulations.

Vendor Quotes

The suppliers' response to "Request for Vendor Information".

SECTION B

GENERIC ARCHITECTURE

ENTERPRISE ACTIVITIES
(ABRIDGED)

3. Manufacture Product (A3)

4. Control Manufactured Product (A4)

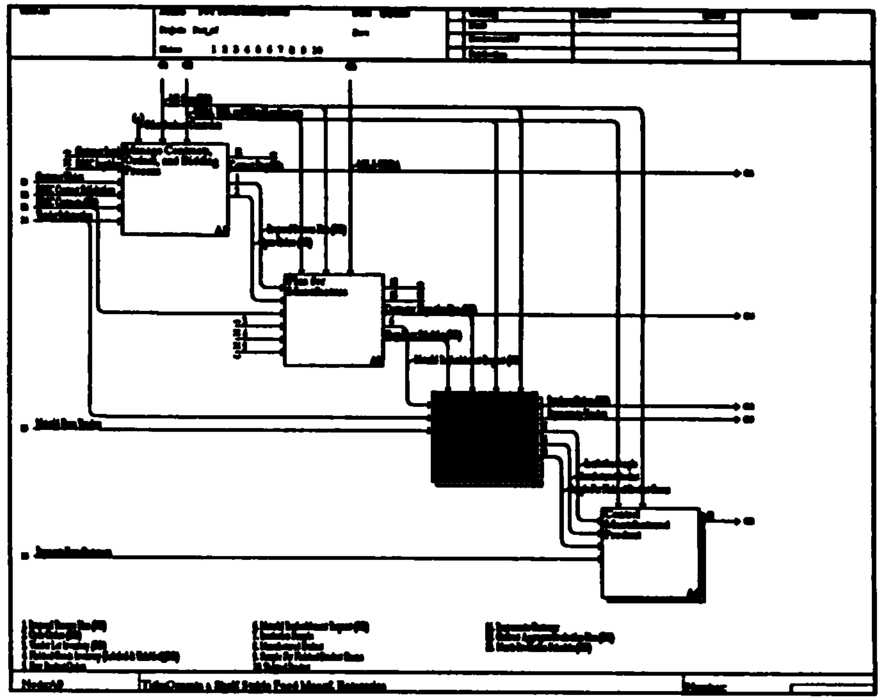
TABLE OF CONTENTS

DIAGRAMS

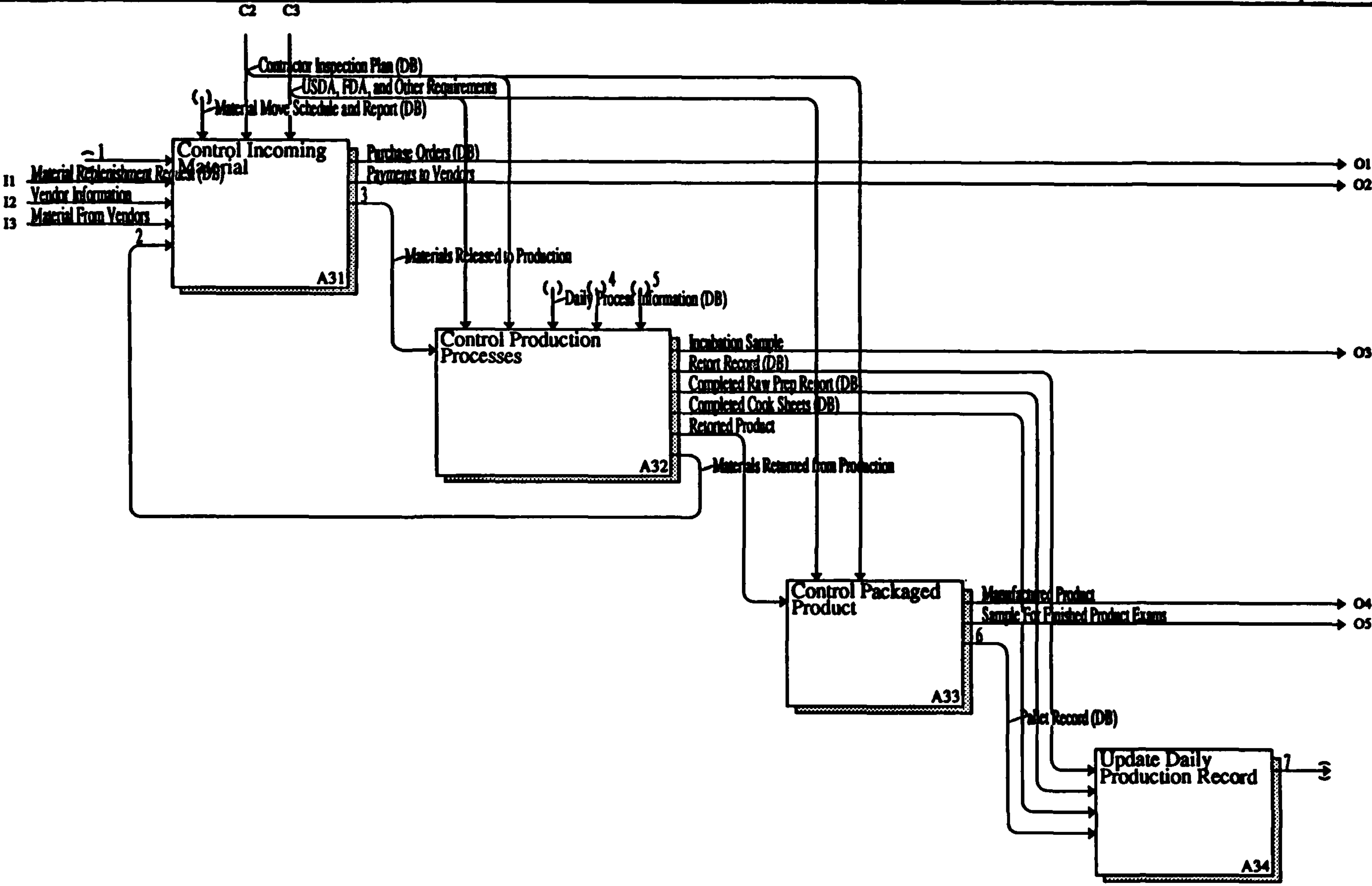
Manufacture Product

This function is comprised of all the activities of the manufacturing facility that transform incoming material into a product. For civilian products, these operations are controlled by USDA, FDA, and other requirements. For combat rations, operations are controlled by the contractor's "Contractor Inspection Plan", which includes statistical process control. The contractor inspection plan incorporates Mil Spec requirements and USDA, FDA, and other requirements.

The "Manufacture Product" function breaks down into four major areas: "Control Incoming Material", "Control Production Processes", "Control Packaged Product", and "Update Daily Production Records". The "Control Incoming Material" function includes all activities from the time material is received until it is released to production. The "Control Production Processes" function includes all activities of manufacturing through retorting. The "Control Packaged Product" function includes inspection, post retort labeling where appropriate, and palletizing. The "Update Daily Production Records" function adds daily production information to the database.



Used At:	Author: PUT YOUR NAME HERE Date: 1/2/1992 Project: Part_of Rev: Notes: 1 2 3 4 5 6 7 8 9 10	Working Draft Recommended Publication	READER DATE



1. Material Requisition (Contracts)
2. Materials Returned from Production
3. Materials Released to Production
4. Cook Sheet (DB)
5. Day Production Schedule (DB)

6. Pallet Record (DB)
7. Updated Database Record (DB)

Node: A3 | Title: Manufacture Product | Number: 29

GLOSSARY

Attached Concepts

Completed Cook Sheets (DB)

The record of the actual batch made for a a specific product, date, shift, and time. Also serves as one of the records of ingredients consumed.

Completed Raw Prep Report (DB)

A record of the actual raw ingredients prepared on a specific day and shift. Also serves as one of the records of ingredients consumed.

Contractor Inspection Plan (DB)

A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and final product inspection.

Cook Sheet (DB)

A description of the ingredients and their quanitities required for a batch of product. Also includes parameters for precooking the batch.

Daily Process Information (DB)

Retort parameters provided to retort operators for particular products to be produced during a production shift.

Day Production Schedule (DB)

The schedule of which products are to be produced on which production equipment for a specific day. This is a firm schedule released the afternoon before the day's production.

Incubation Sample

A sample of product drawn from each retort basket after retorting. Held for ten days and examined for container swelling. Part of the process of insuring that the package contents are properly stetilized.

Manufactured Product

Combat rations or civilian product that have completed the production processes and are awaiting clearence for acceptable quality.

Material From Vendors
Primary raw materials, services, equipment, and supplies converted by or consumed in the manufacturing process.

Material Move Schedule and Report (DB)
A shop floor schedule issued with the "Day Production Schedule" describing what vendor lots (raw material) should be moved from inventory to shop floor locations to accomodate the "Day Production Schedule". Also used to report depletion of raw material.

Material Replenishment Request (DB)
A request to the materials manager for the procurement of material from a supplier to the manufacturing enterprise.

Material Requisition (Contracts)
A request from a contract administrator to purchasing for the ordering or contracting of materials for a specific contract.

Materials Released to Production
Ingredients, pouches, and supplies that are released from inventory to be consumed into work-in-process inventory.

Materials Returned from Production
Ingredients, pouches, and supplies that were released to work-in-process but not consumed during the period of production.

Mil Spec (DB)
The specification for manufacturing the combat ration. Includes ingredients by weight, preparation procedures, quality assurance provisions, and packaging requirements.

Pallet Record (DB)

A record indicating the contents of a pallet, including product and quantities by retort cook and also indicates storage location. Used as a primary record of finished good inventory.

Payments to Vendors
Payments made for the receipt of materials and services from vendors.

Purchase Orders (DB)
A document that initiates the sale of material from a vendor to the manufacturing enterprise.

Retort Record (DB)
A log of the actual retort operating conditions by retort cook. Includes the initial temperature of the product placed in the retort.

Retorted Product
The product after retorting but before palletizing.

Sample For Finished Product Exams
A sample of product taken from production and used for examining the quality of the finished product. Includes a container evaluation and a product sample.

Shop Floor Schelules (DB)
A group of schedules that are provided daily to the line supervisor for executing production. They include the "Day Production Schedule", "Material Move Schedule and Report", "Raw Preparation Sheet", "Cook Sheet", and "Daily Process Information".

Updated Database Record (DB)
Entry of current information into the database. Includes information from retort record, completed raw prep report, and completed cook sheet.

USDA, FDA, and Other Requirements
Includes requirements imposed on the manufacture of food products by federal and

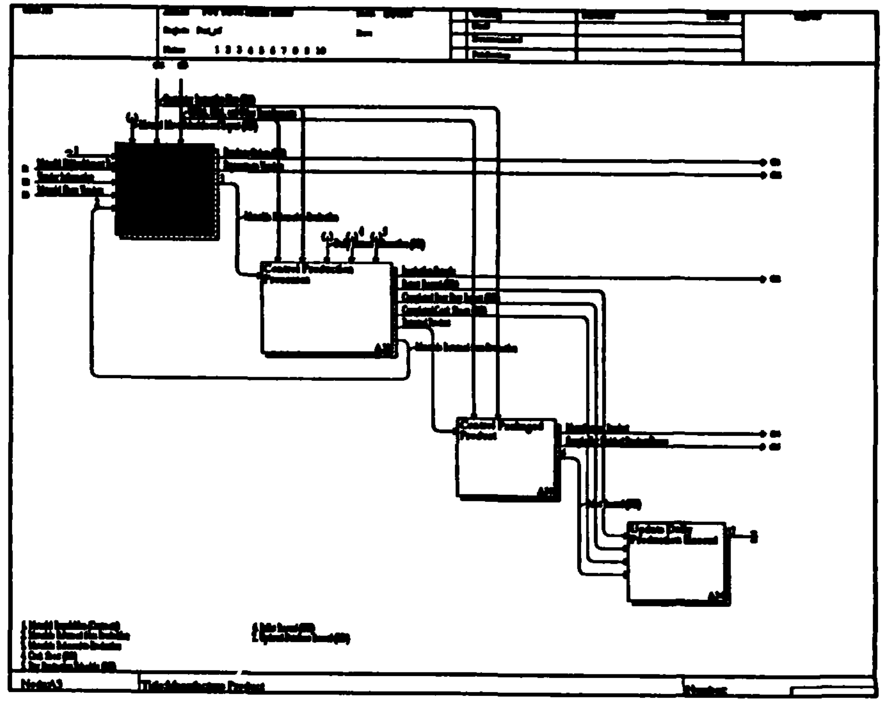
state agencies, for example, the USDA meat and poultry inspection regulations.

Vendor Information

Responses to requests for vendor information.

Control Incoming Material

This function includes the processes by which raw material inventory is replenished and accounted for, as well as the processes by which material is physically handled and stored. The activities begin when the replenishment policy used by the inventory manager triggers a "Material Requisition". This material requisition causes purchasing to "Prepare and Issue Purchase Order". When shipments of materials arrive, they are handled by the "Receive, Inspect, and Store" function. Finally, the acceptance of a shipment of materials triggers a series of accounting transactions in the "Update Accounting and Authorize Payment" function.



Used At:

Author: PUT YOUR NAME HERE Date: 1/2/1992

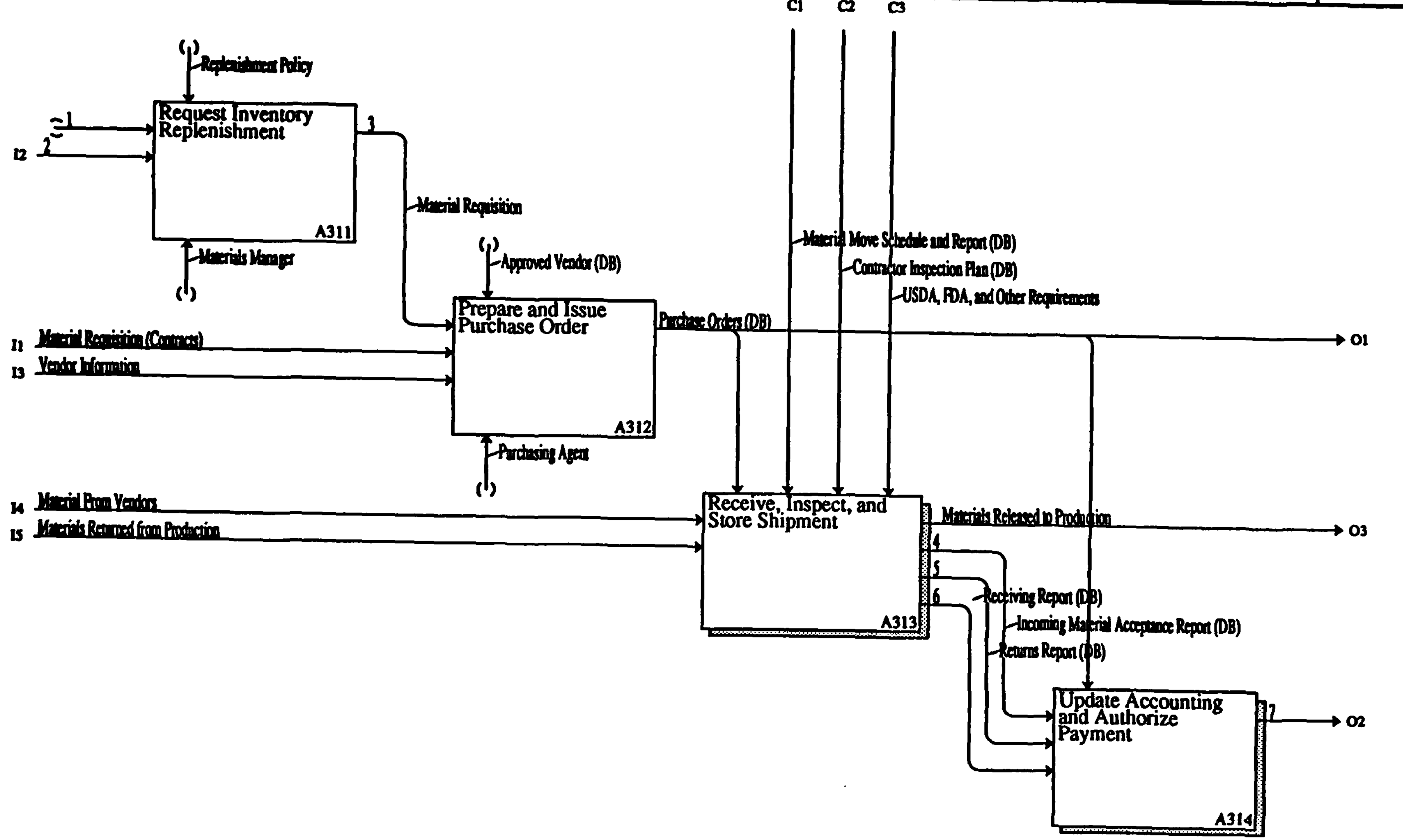
Project: Part_of Rev:

Notes: 1 2 3 4 5 6 7 8 9 10

Working
Draft
Recommended
Publication

READER DATE

Context



1. Inventory Position (DB)
2. Material Replenishment Request (DB)
3. Material Requisition
4. Incoming Material Acceptance Report (DB)
5. Materials Returned to Suppliers
6. Receiving Report (DB)
7. Payments to Vendors

Node:A31 Title:Control Incoming Material Number: 35

GLOSSARY

Attached Concepts

Approved Vendor (DB)

An enterprise database that lists vendors that are approved sources of ingredients and materials.

Contractor Inspection Plan (DB)

A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and final product inspection.

Incoming Material Acceptance Report (DB)

Reports on the inspection of incoming materials using testing procedures administered by quality assurance in conformance with the Contractor Inspection Plan.

Inventory Position (DB)

A database attribute that monitors the number of units on-hand and on-order of materials and ingredients used in production.

Material From Vendors

Primary raw materials, services, equipment, and supplies converted by or consumed in the manufacturing process.

Material Move Schedule and Report (DB)

A shop floor schedule issued with the "Day Production Schedule" describing what vendor lots (raw material) should be moved from inventory to shop floor locations to accomodate the "Day Production Schedule". Also used to report depletion of raw material.

Material Replenishment Request (DB)

A request to the materials manager for the procurement of material from a supplier to the manufacturing enterprise.

Material Requisition

A request from production planning to the materials manager to provide materials needed for future production.

Material Requisition (Contracts)
A request from a contract administrator to purchasing for the ordering or contracting of materials for a specific contract.

Materials Manager
The individual who is required to maintain the overall operation of the in-plant materials supply and storage.

Materials Released to Production
Ingredients, pouches, and supplies that are released from inventory to be consumed into work-in-process inventory.

Materials Returned from Production
Ingredients, pouches, and supplies that were released to work-in-process but not consumed during the period of production.

Materials Returned to Suppliers
If Received materials do not pass inspection, they are returned to suppliers for credit.

Payments to Vendors
Payments made for the receipt of materials and services from vendors.

Purchase Orders (DB)
A document that initiates the sale of material from a vendor to the manufacturing enterprise.

Purchasing Agent
The individual responsible for buying material and supplies for the enterprise.

Receiving Report (DB)

A report from the receiving department indicating the materials received against purchase orders during each day.

Replenishment Policy

The rules that are used to determine when to order materials from suppliers and how much to order.

Returns Report (DB)

A report to notify accounting that material previously received from a vendor has been returned to the vendor because it was non-conforming.

USDA, FDA, and Other Requirements

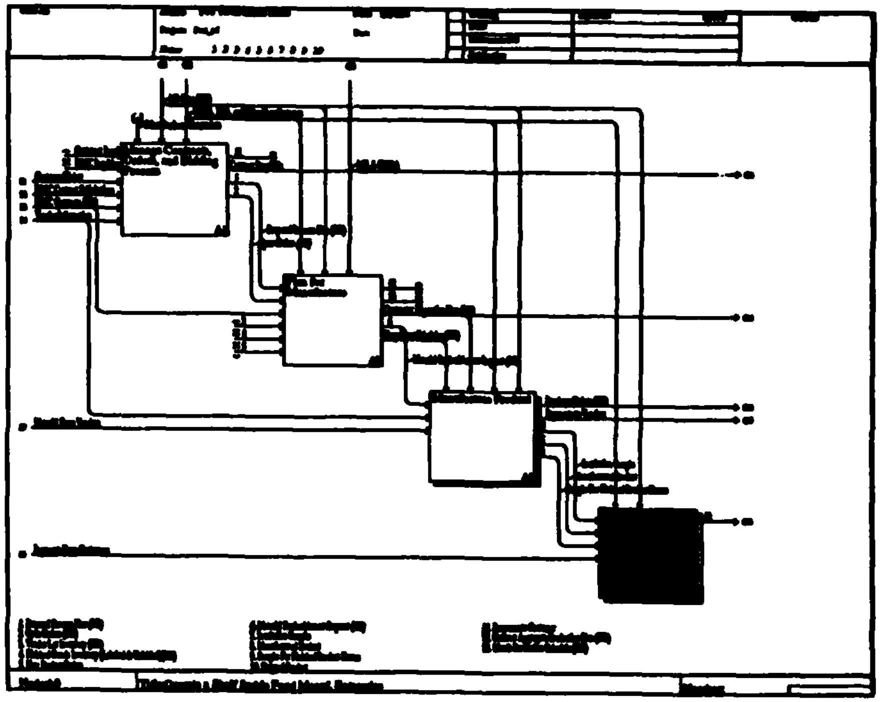
Includes requirements imposed on the manufacture of food products by federal and state agencies, for example, the USDA meat and poultry inspection regulations.

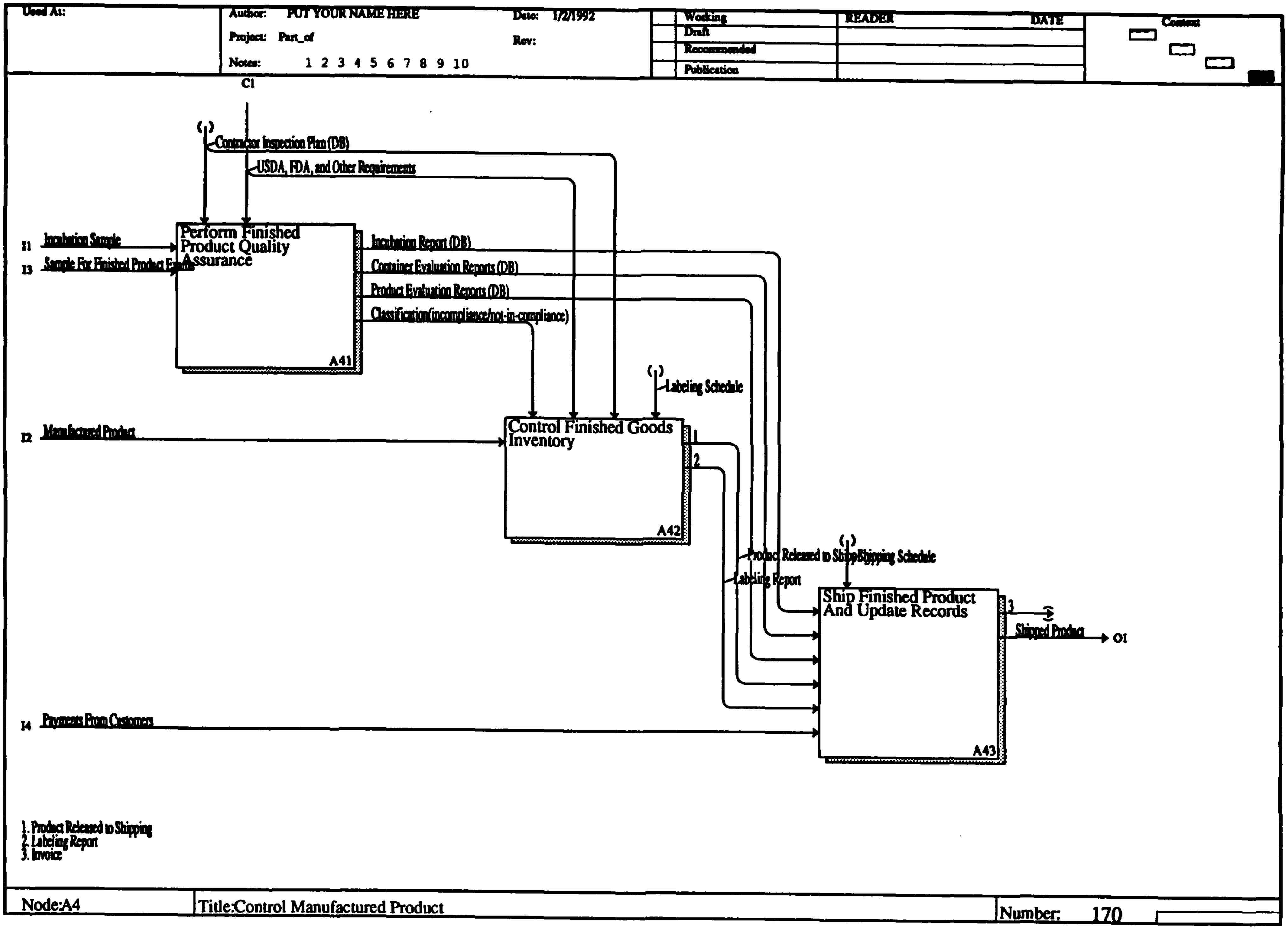
Vendor Information

Responses to requests for vendor information.

Control Manufactured Product

This includes activities that occur after container... -...ssed and leave the production area to go to labeling or to be inventoried. Subactivities are: "Perform Finished Product Quality Assurance", "Control Finished Goods Inventory", and "Ship finished Product and Update Records".




Used At:
Author: PUT YOUR NAME HERE
Date: 1/2/1992
Project: Part_of
Rev:
Notes: 1 2 3 4 5 6 7 8 9 10
Working
Draft
Recommended
Publication
READER
DATE
Context
C1
Contractor Inspection Plan (DB)
USDA, FDA, and Other Requirements
I1 Incubation Sample
I3 Sample For Finished Product Evals
Perform Finished Product Quality Assurance
A41
Incubation Report (DB)
Container Evaluation Reports (DB)
Product Evaluation Reports (DB)
Classification(incompliance/not-in-compliance)
Labeling Schedule
I2 Manufactured Product
Control Finished Goods Inventory
A42
1
2
Product Released to Ship
Shipping Schedule
Labeling Report
Ship Finished Product And Update Records
A43
3
Shipped Product
O1
I4 Payments From Customers
1. Product Released to Shipping
2. Labeling Report
3. Invoice
Node:A4
Title:Control Manufactured Product
Number: 170

GLOSSARY

Attached Concepts

Classification(incompliance/not-in-compliance)
The classification given to finished goods by quality assurance.

Container Evaluation Reports (DB)
An examination of a sample of finished product for seal integrity.

Contractor Inspection Plan (DB)
A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and final product inspection.

Incubation Report (DB)
The result of the evaluation of the incubation sample.

Incubation Sample
A sample of product drawn from each retort basket after retorting. Held for ten days and examined for container swelling. Part of the process of insuring that the package contents are properly stetilized.

Invoice
A request to a customer for payment for shipped product.

Labeling Report
A report from the labeling department of which pallets were labeled from finished goods inventory or from the day's production.

Labeling Schedule
Daily instrucations to the labeling department indicating which products are to be labeled from daily production and/or from inventory.

Manufactured Product

Combat rations or civilian product that have completed the production processes and are awaiting clearence for acceptable quality.

Mil Spec (DB)
The specification for manufacturing the combat ration. Includes ingredients by weight, preparation procedures, quality assurance provisions, and packaging requirements.

Payments From Customers
Payments received for the delivery of shipped product.

Product Evaluation Reports (DB)
Report of the finish product examination for conformance with specifications.

Product Released to Shipping
Product to be staged at shipping dock for pickup and delivery to customers.

Sample For Finished Product Exams
A sample of product taken from production and used for examining the quality of the finished product. Includes a container evaluation and a product sample.

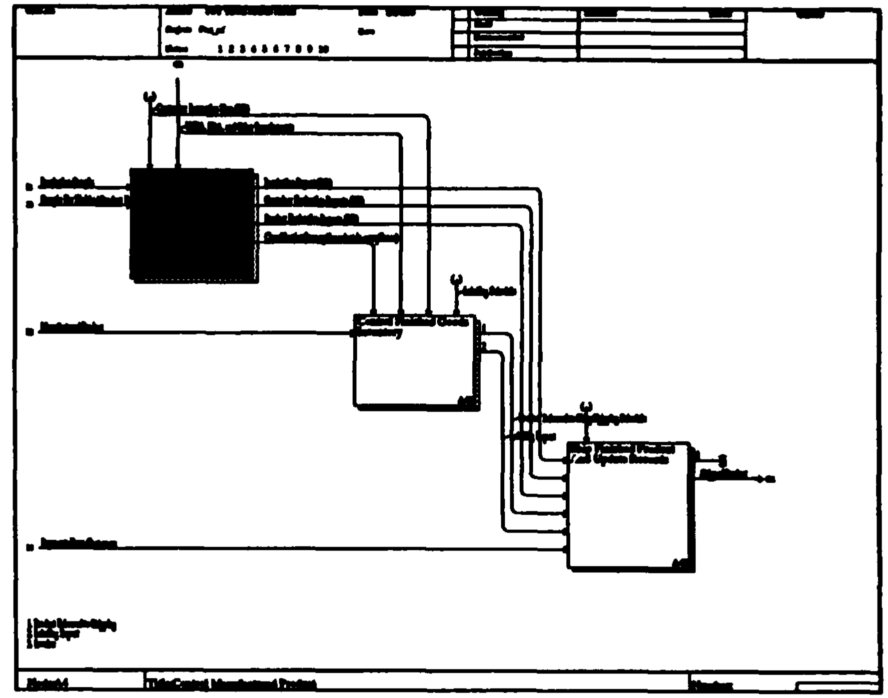
Shipped Product
Rations or civilian products produced to specification and shipped to appointed destination.

Shipping Schedule
A schedule indicating the pallets of product that are to be shipped to specific customers on a specific day.

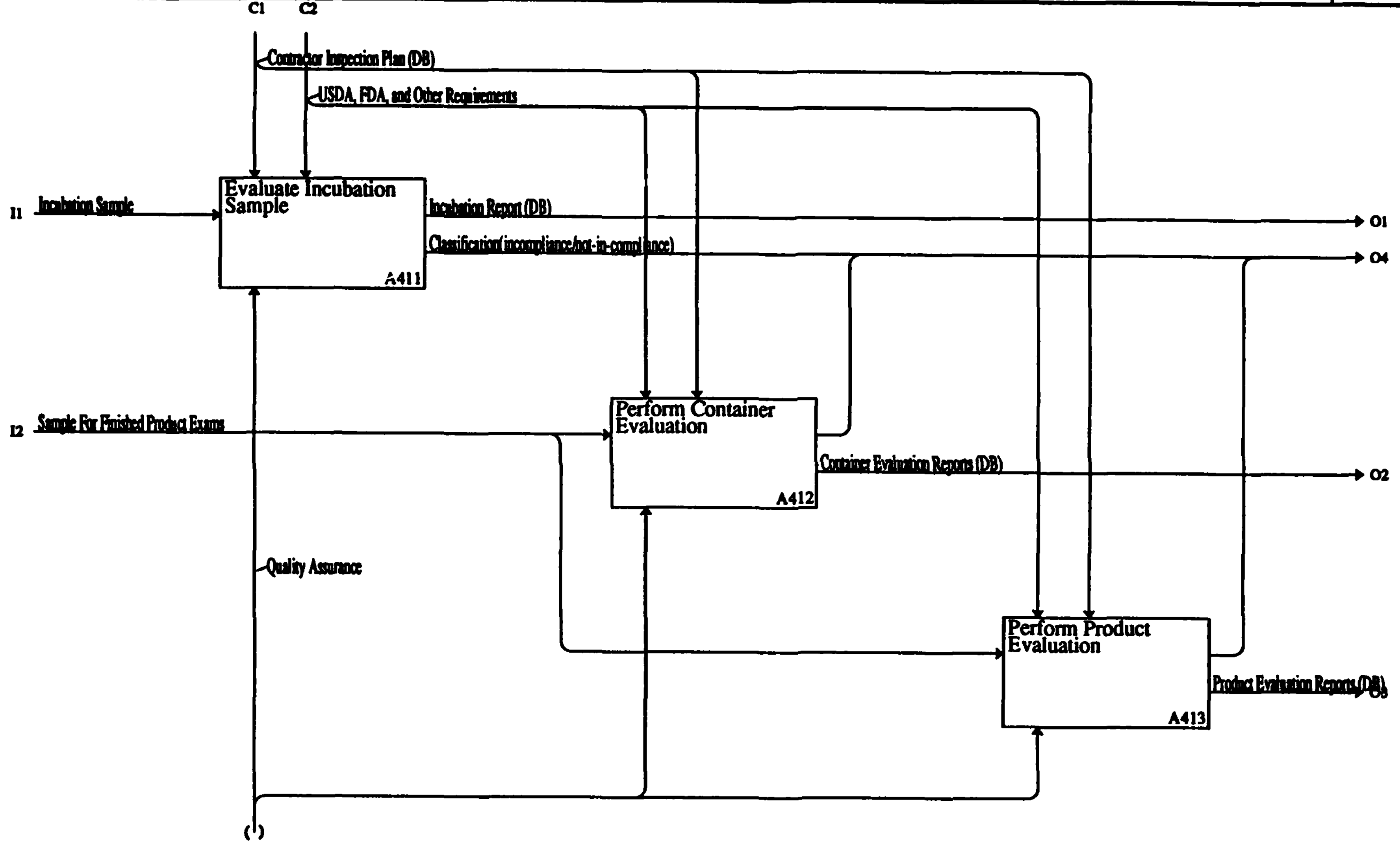
USDA, FDA, and Other Requirements
Includes requirements imposed on the manufacture of food products by federal and state agencies, for example, the USDA meat and poultry inspection regulations.

Perform Finished Product Quality Assurance

Samples of finished product are taken from the production line after retorting to undergo finished product examinations. There are three classes of finished product examination as given by the three subactivities: "Evaluate Incubation Sample", "Perform Container Evaluation", and "Perform Product Evaluation".



Used At:	Author: PUT YOUR NAME HERE	Date: 1/2/1992	Working	READER	DATE	Context:
	Project: Part_of	Rev:	Draft			
	Notes: 1 2 3 4 5 6 7 8 9 10		Recommended			
			Publication			



Node:A41	Title:Perform Finished Product Quality Assurance	Number: 174

GLOSSARY

Attached Concepts

Classification(incompliance/not-in-compliance)
The classification given to finished goods by quality assurance.

Container Evaluation Reports (DB)
An examination of a sample of finished product for seal integrity.

Contractor Inspection Plan (DB)
A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and final product inspection.

Incubation Report (DB)
The result of the evaluation of the incubation sample.

Incubation Sample
A sample of product drawn from each retort basket after retorting. Held for ten days and examined for container swelling. Part of the process of insuring that the package contents are properly stetilized.

Product Evaluation Reports (DB)
Report of the finish product examination for conformance with specifications.

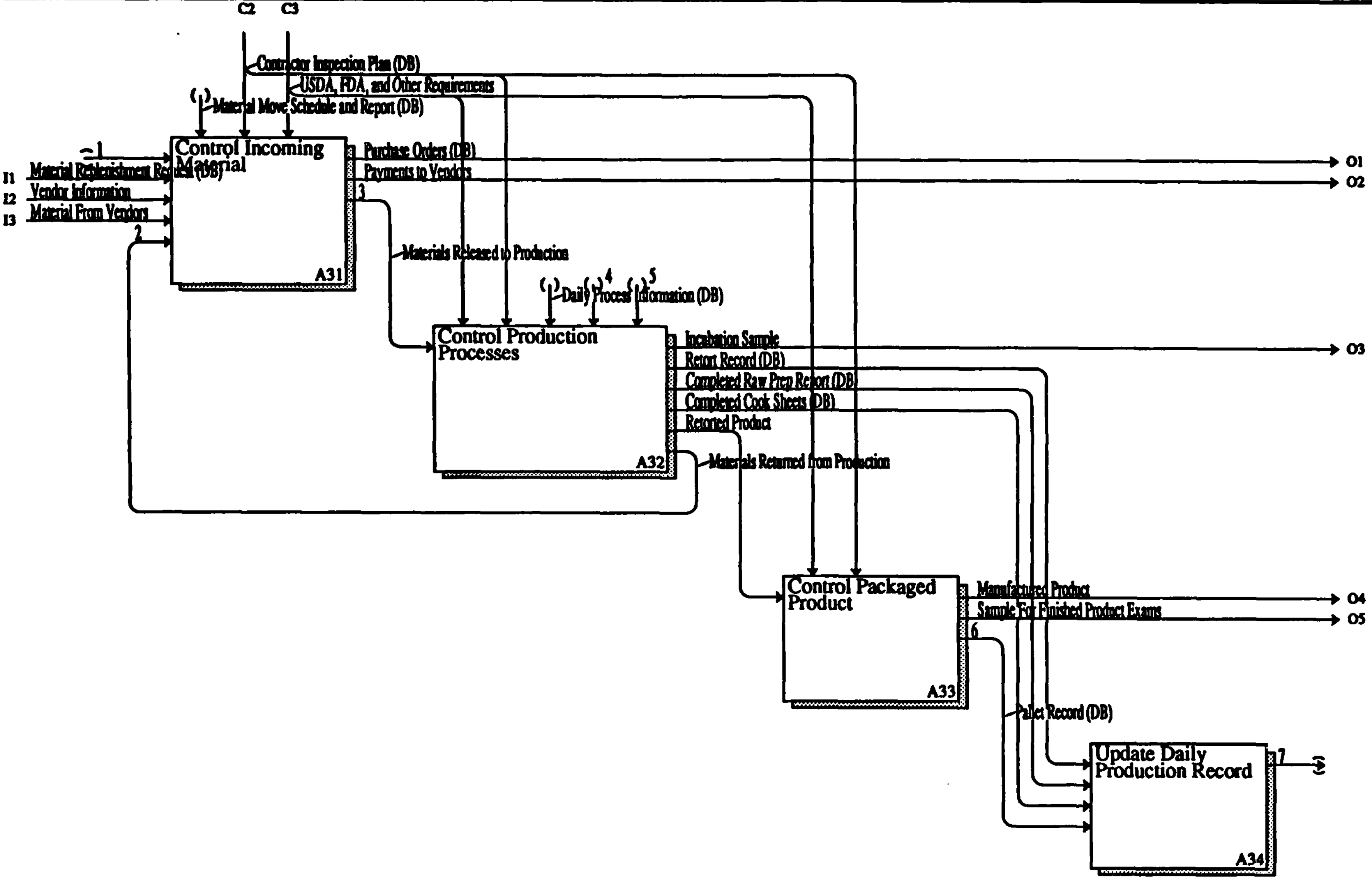
Quality Assurance
Department with overall responsibility for ingredient, material, and product testing to insure conformity to specifications.

Sample For Finished Product Exams
A sample of product taken from production and used for examining the quality of the finished product. Includes a container evaluation and a product sample.

USDA, FDA, and Other Requirements

Includes requirements imposed on the manufacture of food products by federal and state agencies, for example, the USDA meat and poultry inspection regulations.

Used At:	Author: PUT YOUR NAME HERE	Date: 1/2/1992	Working	READER	DATE	Context
	Project: Part_of	Rev:	Draft			
	Notes: 1 2 3 4 5 6 7 8 9 10		Recommended			
			Publication			



1. Material Requisition (Contracts)
2. Materials Returned from Production
3. Materials Released to Production
4. Cook Sheet (DB)
5. Day Production Schedule (DB)
6. Pallet Record (DB)
7. Updated Database Record (DB)

Node: A3	Title: Manufacture Product	Number: 29

APPENDIX I

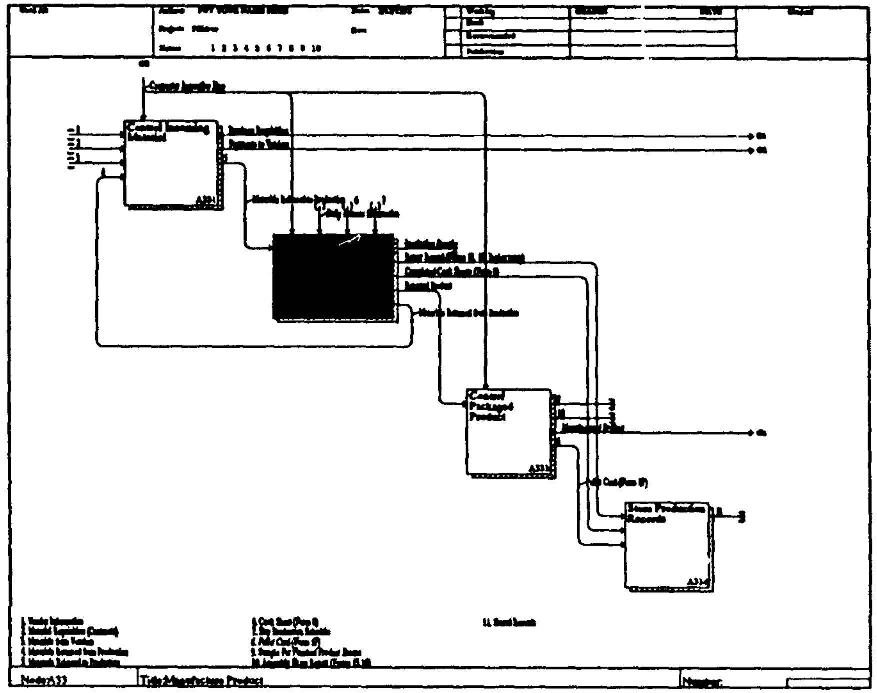
MRE POUCH - OMELET AND HAM

CASE STUDY
(ABRIDGED)

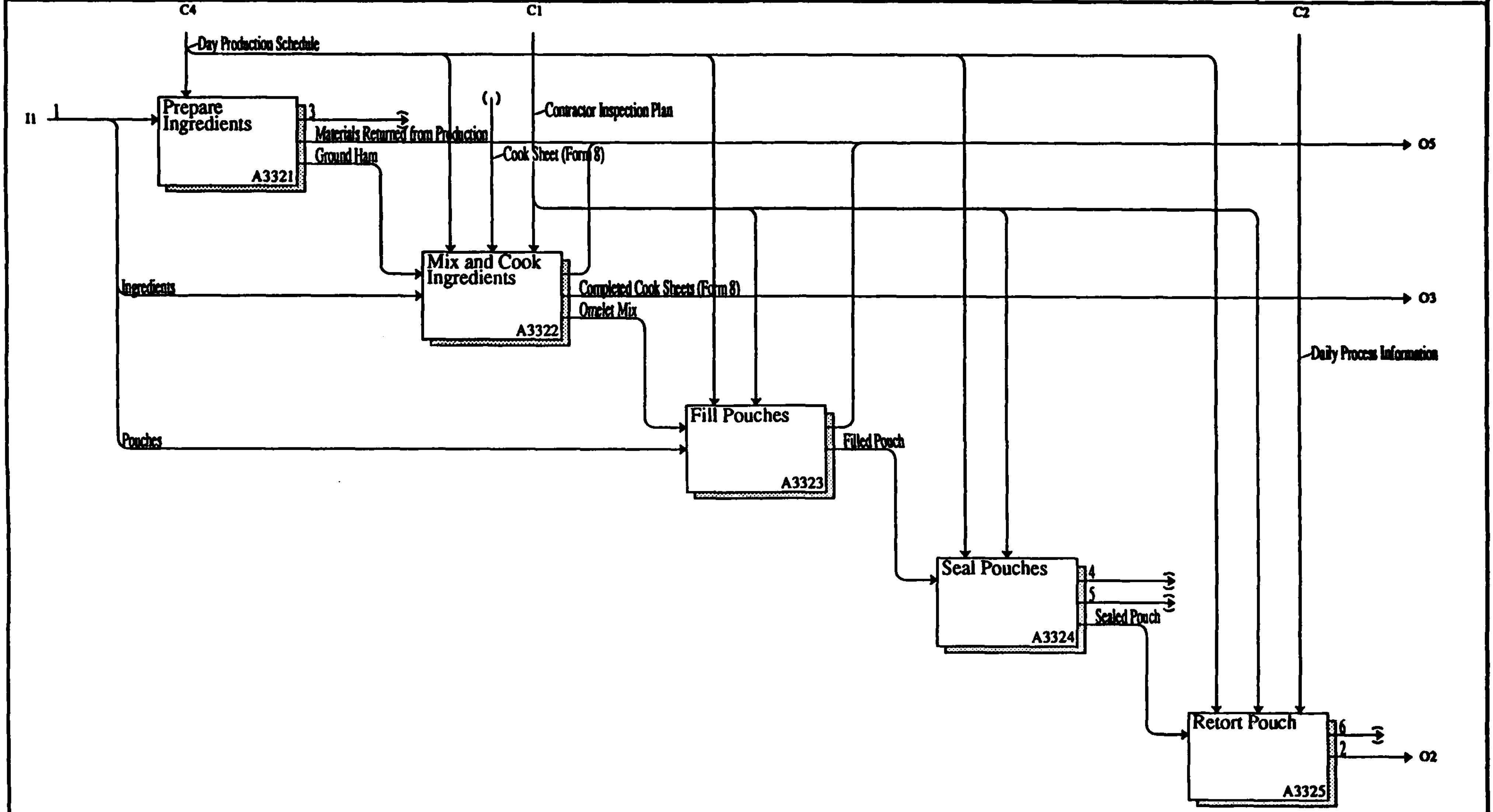
TABLE OF CONTENTS

Control Production Processes

This section describes the production of Ham Omlet in the MRE pouch. It covers the activities that begin when material is released to work-in-process and ends when product has completed the retort operation. The subactivities include "Prepare Ingredients", "Mix and Cook Ingredients", "Fill Pouches", "Seal Pouches", and "Retort Product".



Used At:	Author: PUT YOUR NAME HERE	Date: 2/12/1992	Working	READER	DATE	Context
	Project: Pillsbur	Rev:	Draft			
	Notes: 1 2 3 4 5 6 7 8 9 10		Recommended			
			Publication			



1. Materials Released to Production
2. Retort Record (Forms 12, 13, Taylor temp)
3. Completed Ham Usage Report (Form 6)
4. SS/RG Report (Form 9)
5. Weight Report (Form 10)

6. Retorted Omelet

Node:A332	Title:Control Production Processes	Number: 77

GLOSSARY

Attached Concepts

Completed Cook Sheets (Form 8)
The record of actual batch made for a specific product, date, shift, and time.

Completed Ham Usage Report (Form 6)
A report of the quantity of ham consumed in production during a shift.

Contractor Inspection Plan
A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and final product inspection.

Cook Sheet (Form 8)
A description of the ingredients and their quanitities required for a batch of product. Also includes paramaters for precooking the batch.

Cook Sheet (Form 8)
A description of the ingredients and their quantities required for a batch of product. Also includes parameters for precooking the batch.

Daily Process Information
Retort paramaters provided to the retort operator for particular products to be produced during a production shift.

Day Production Schedule
The schedule of which products are to be produced on which production equipment for a specific day. This is a firm schedule released the afternoon before the day's production.

Filled Pouch
Pouch containing product but not sealed.

Ground Ham

Ham that is ground at a step prior to mixing and precooking.

Incubation Sample
A sample of product drawn from each retort basket after retorting. Held for ten days and examined for container swelling. Part of the process of insuring that the package contents are properly sterilized.

Ingredients
Components used to produce a food product.

Materials Released to Production
Ingredients, pouches, and supplies that are released from inventory to be consumed into work-in-process inventory.

Materials Returned from Production
Ingredients, pouches, and supplies that were released to work-in-process but not consumed during the period of production.

Omelet Mix
Mixture of eggs, ham, hominy grits, and spices after precooking but before filling.

Pouches
Flexible packaging material.

Retort Record (Forms 12, 13, Taylor temp)
A set of logs of the actual retort operating conditions by retort cook. Includes the initial temperature of the product placed in the retort.

Retorted Omelet
The omelet after retorting but before palletizing.

Retorted Product
The product after retorting but before palletizing.

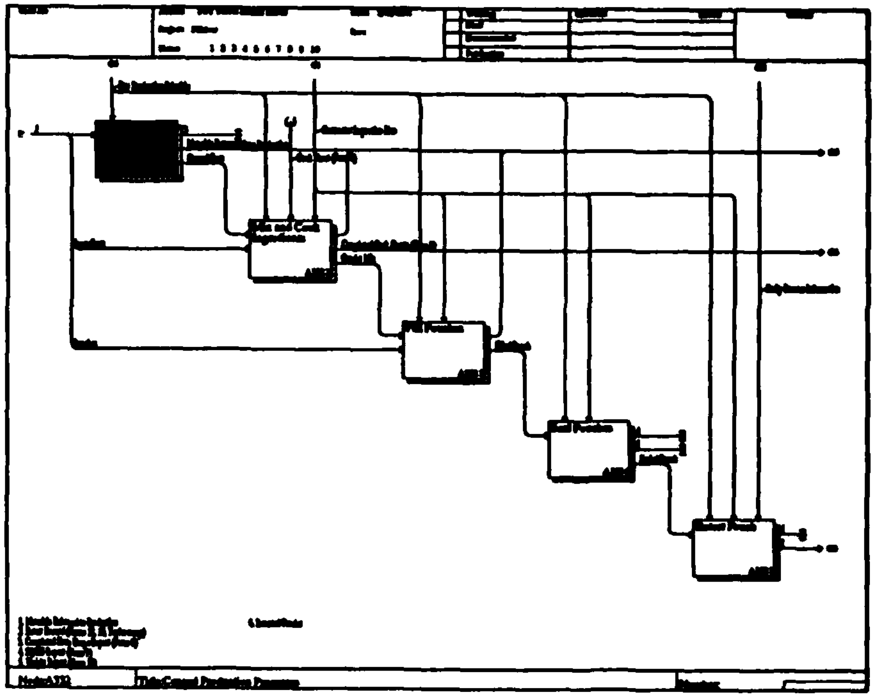
Sealed Pouch
A pouch after leaving the sealing operation but prior to retorting.

SS/RG Report (Form 9)
A report on seal strength and residual gas for a sample of sealed pouches.

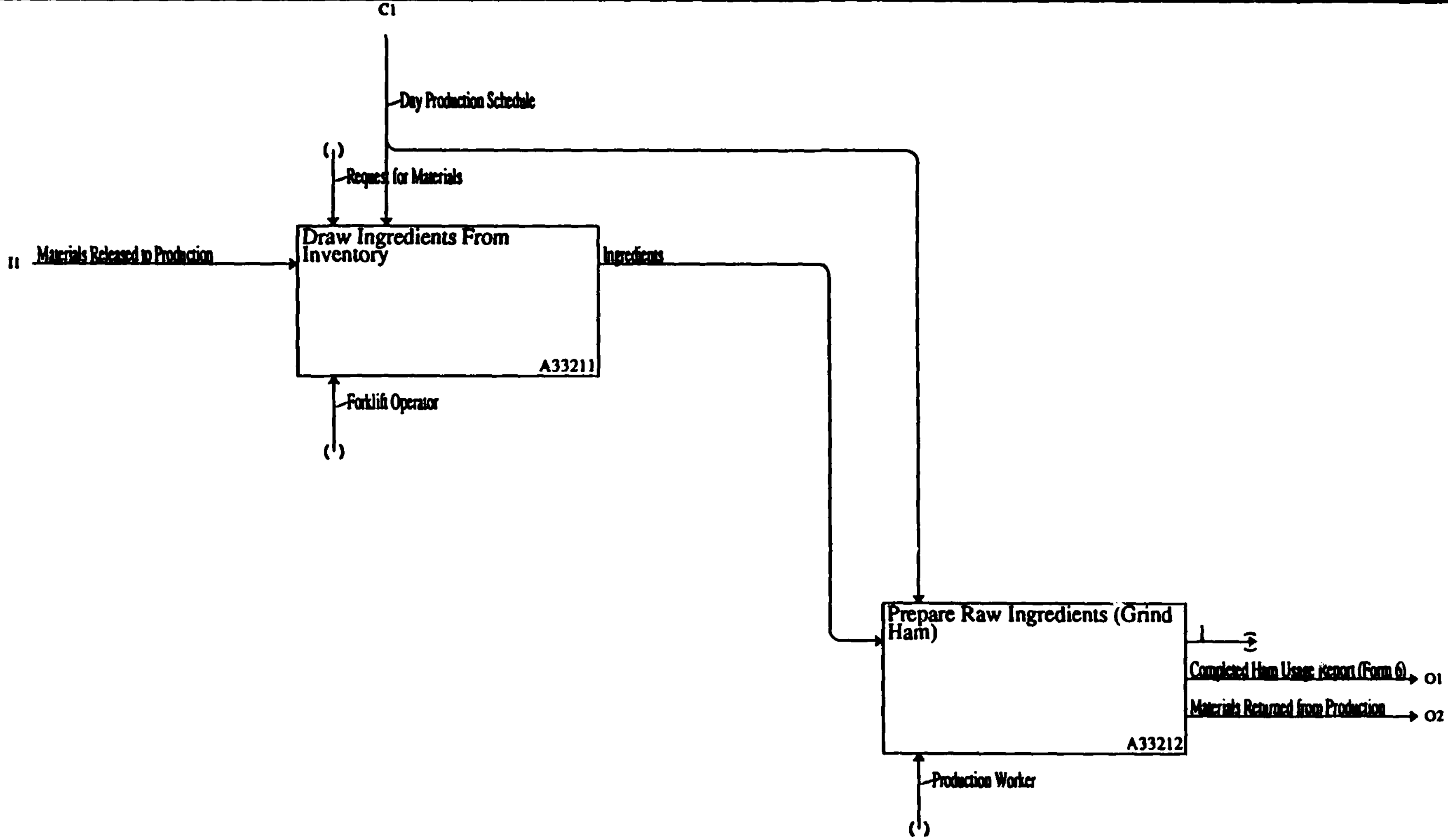
Weight Report (Form 10)
A quality control chart that tracks the fill weights of sealed packages.

Prepare Ingredients

This activity provides "Prepared Ingredients (Ground Ham)" for the mixing and cooking operations. There are two subactivities: "Draw Ingredients From Inventory", and "Prepare Raw Ingredients (Grind Ham)".



Used At:	Author: PUT YOUR NAME HERE	Date: 2/12/1992	Working	READER	DATE	Context
	Project: Pillsbur	Rev:	Draft			
	Notes: 1 2 3 4 5 6 7 8 9 10		Recommended			
			Publication			



1. Prepared Ingredients (Ground Ham)

Node:A3321	Title:Prepare Ingredients	Number: 82

GLOSSARY

Attached Concepts

Completed Ham Usage Report (Form 6)

A report of the quantity of ham consumed in production during a shift.

Day Production Schedule

The schedule of which products are to be produced on which production equipment for a specific day. This is a firm schedule released the afternoon before the day's production.

Forklift Operator

A person who performs material handling using a forklift truck.

Ground Ham

Ham that is ground at a step prior to mixing and precooking.

Ingredients

Components used to produce a food product.

Materials Released to Production

Ingredients, pouches, and supplies that are released from inventory to be consumed into work-in-process inventory.

Materials Returned from Production

Ingredients, pouches, and supplies that were released to work-in-process but not consumed during the period of production.

Prepared Ingredients (Ground Ham)

Ham that has been ground prior to mixing and precooking.

Production Worker

Individual employed by the enterprise as production labor.

Request for Materials

eee A request by production to release ingredients into work-in-process.

APPENDIX II

TRAY PACK - BEEF CHUNKS AND GRAVY

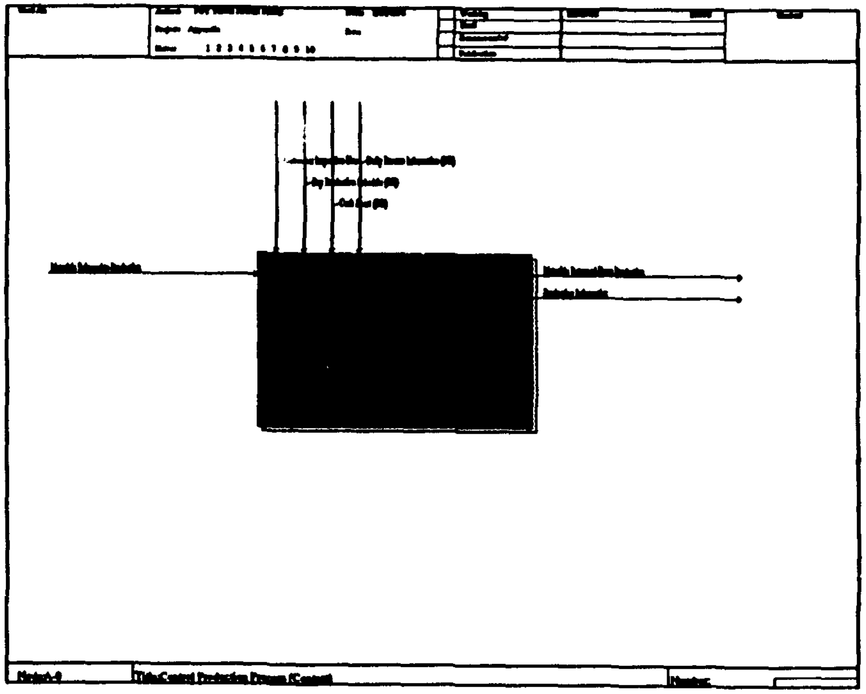
CASE STUDY
(ABRIDGED)

TABLE OF CONTENTS

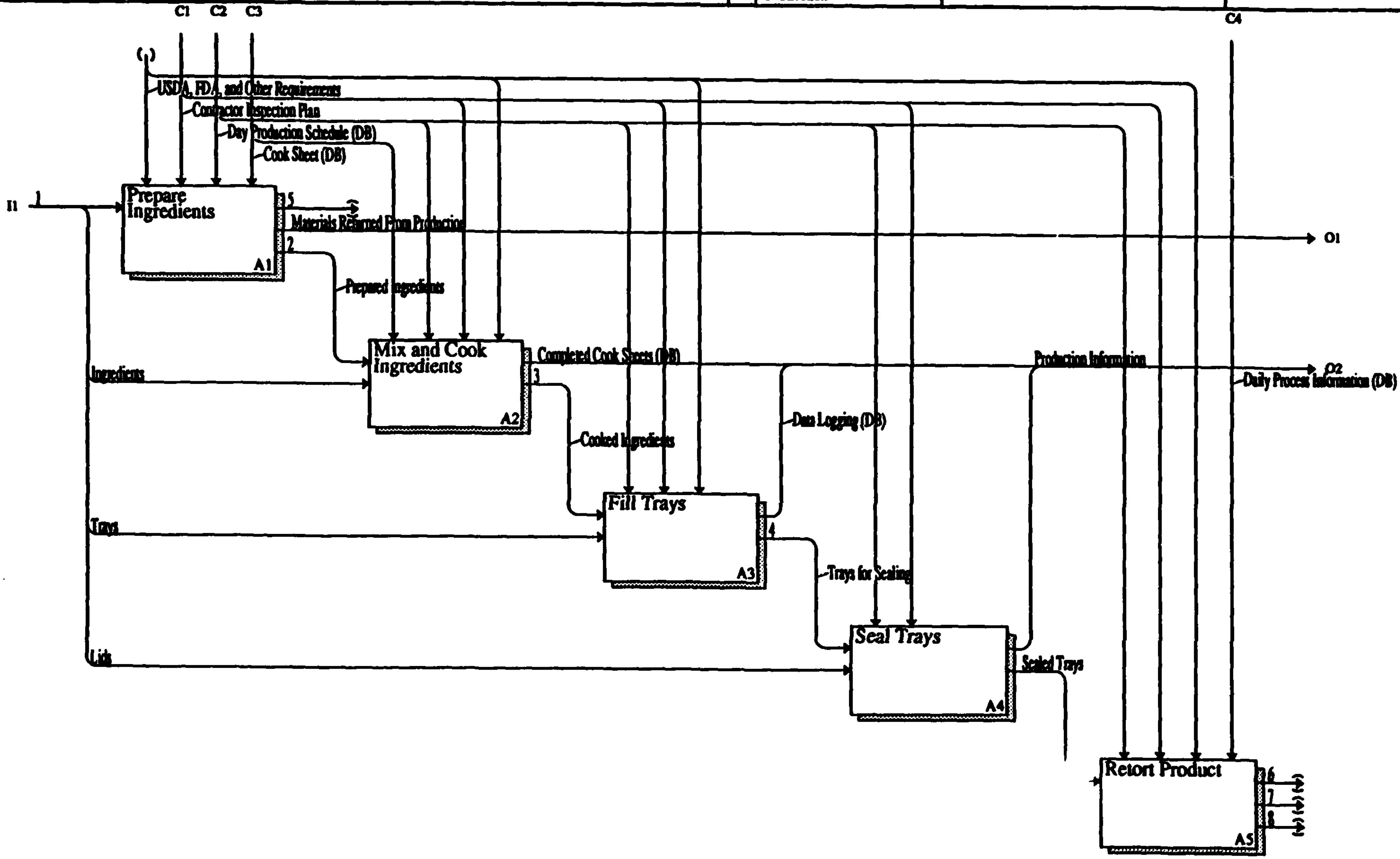
Control Production Process

This case describes the routing of material in a highly automated factory. The activity illustrates the case of beef chunks in gravy in which cooking of meats is done separately in an air impingement oven. Material handling, filling, sealing, retorting, and testing are automated and computer integrated.

There are five subactivities: "Prepare Ingredients", "Mix and Cook Ingredients", "Fill Trays", "Seal Trays", and "Retort Product". The product is being packaged on a traypack line.



Used At:	Author: PUT YOUR NAME HERE	Date: 1/27/1992	Working	READER	DATE	Context
	Project: Appendix	Rev:	Draft			
	Notes: 1 2 3 4 5 6 7 8 9 10		Recommended			
			Publication			



1. Materials Released to Production
2. Prepared Ingredients
3. Cooked Ingredients
4. Trays for Sealing
5. Completed Subassembly Report (DB)

6. Incubation Sample
7. Retort Record (DB)
8. Retorted Product

Node: A0	Title: Control Production Process	Number: 7

GLOSSARY

Attached Concepts

Completed Cook Sheets (DB)

The record of the actual raw ingredients prepared on a specific day and shift. Also serves as one of the records of ingredients consumed.

Completed Subassembly Report (DB)

A report which indicates the number of subassemblies, typically spice mixes, made during a production shift in conformance with the cook sheet specification.

Contractor Inspection Plan

A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and the final inspection.

Cook Sheet (DB)

A description of the ingredients and their quantities required for a batch of product. Also includes parameters for precooking the batch.

Cooked Ingredients

Ingredients after precooking but before filling.

Daily Process Information (DB)

Retort parameters provided to retort operators for particular products to be produced during a production shift.

Data Logging (DB)

Automatic storage of data from sensors to PLC and, subsequently, to factory database.

Day Production Schedule (DB)

The schedule of which products are to be produced on which production equipment for a specific day. This is a firm schedule released the afternoon before the day's production.

Incubation Sample

A sample of product drawn from each retort basket after retorting. Held for ten days and examined for container swelling. Part of the process of insuring that the package contents are properly sealed and sterilized.

Ingredients
Components used to produce a food product.

Lids
Covers to be sealed on tray-cans.

Materials Released to Production
Ingredients, trays, and supplies that are released from inventory to be consumed into work-in-process inventory.

Materials Returned From Production
Ingredients, trays, and supplies that were released into work-in-process but not consumed during the period of production.

Prepared Ingredients
Ingredients that are combined into a subassembly at a step prior to mixing and precooking.

Retort Record (DB)
A log of the actual retort operating conditions by retort cook. Includes the initial temperature of the product placed in the retort.

Retorted Product
The product after retorting but before palletizing.

Sealed Trays
A tray after leaving the sealing operation but prior to retorting

Trays

Traycans to be filled.

Trays for Sealing

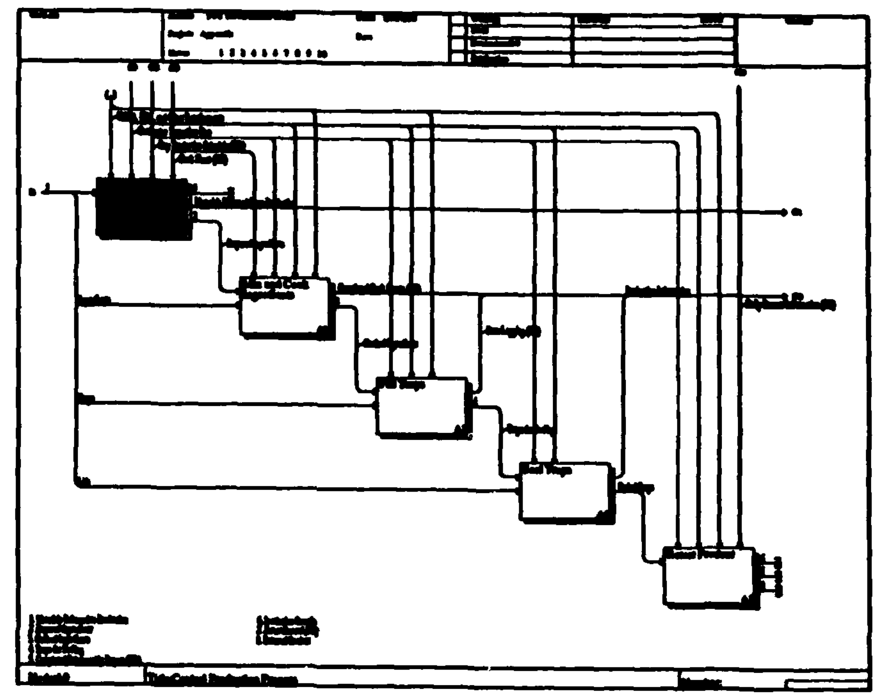
Trays that have been filled and are ready for sealing.

USDA, FDA, and Other Requirements

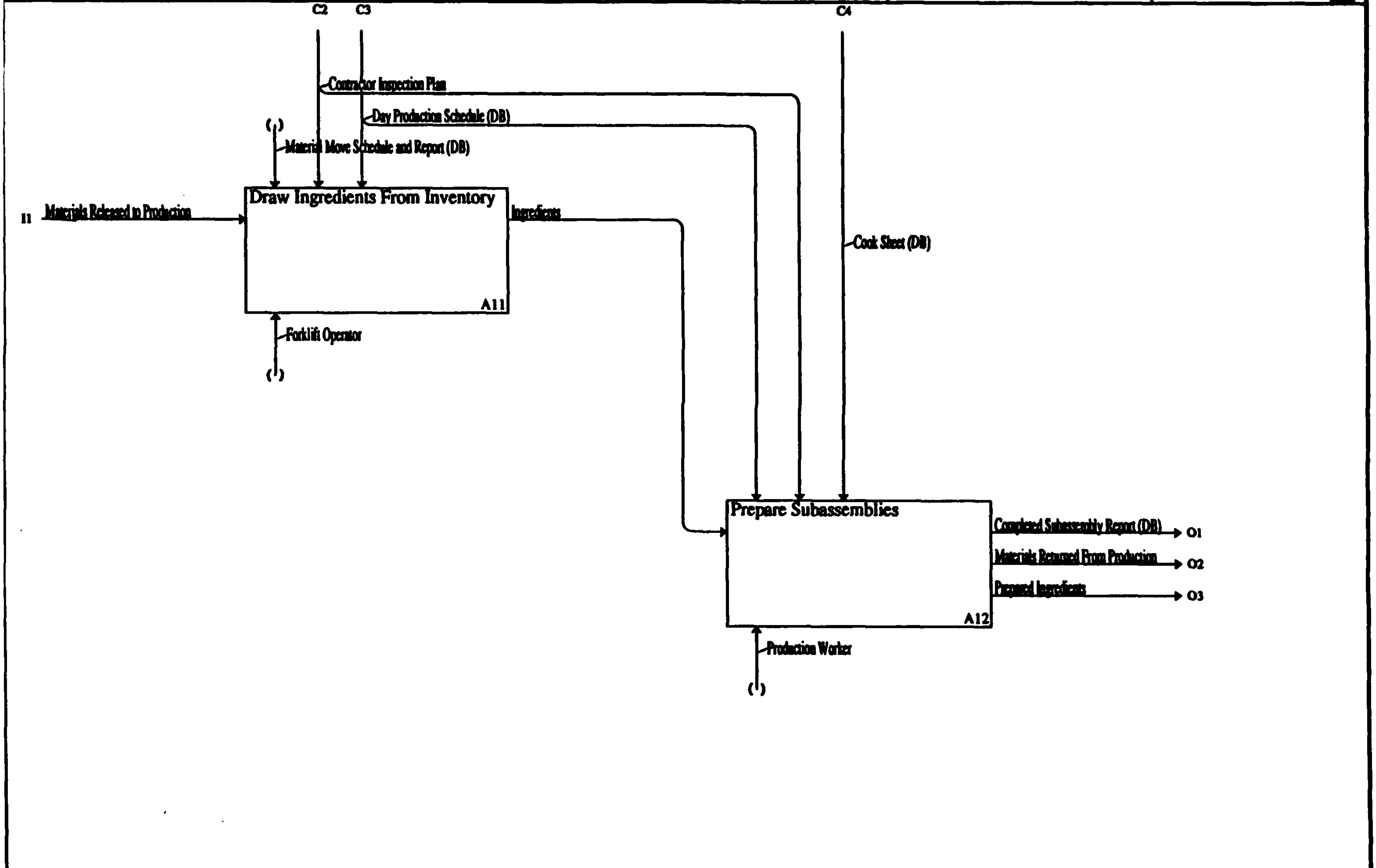
Includes requirements imposed on the manufacture of food products by federal and state agencies, for example, the USDA meat and poultry inspection regulations.

Prepare Ingredients

This activity provides prepared ingredients and subassemblies for the mixing and cooking operations. There are two subactivities: "Draw Ingredients from Inventory" and "Prepare Subassemblies". The subassembly is a spice mix that is used in the sauce preparation.



Used At:	Author: PUT YOUR NAME HERE	Date: 1/27/1992	Working	READER	DATE	Context
	Project: Appendix	Rev:	Draft			
	Notes: 1 2 3 4 5 6 7 8 9 10		Recommended			
			Publication			



Node: A1	Title: Prepare Ingredients	Number: 12

GLOSSARY

Attached Concepts

Completed Subassembly Report (DB)

A report which indicates the number of subassemblies, typically spice mixes, made during a production shift in conformance with the cook sheet specification.

Contractor Inspection Plan

A document written by the contractor in conformance with MIL-I-45208A. It contains all the contractor's procedures in assuring the quality of foodstuffs offered for sale to the government. It includes indicators to assure the quality and inspection of raw material, the calibration of instruments used in inspection, the quality assurance organization, in-process inspection, and the final inspection.

Cook Sheet (DB)

A description of the ingredients and their quantities required for a batch of product. Also includes parameters for precooking the batch.

Day Production Schedule (DB)

The schedule of which products are to be produced on which production equipment for a specific day. This is a firm schedule released the afternoon before the day's production.

Forklift Operator

A person who performs material handling using a forklift.

Ingredients

Components used to produce a food product.

Material Move Schedule and Report (DB)

A shop floor schedule issued with the "Day Production Schedule" describing what vendor lots (raw material) should be moved from inventory to shop floor locations to accomodate the "Day Production Schedule". Also used to report depletion of raw material.

Materials Released to Production

Ingredients, trays, and supplies that are released from inventory to be consumed into

work-in-process inventory.

Materials Returned From Production

Ingredients, trays, and supplies that were released into work-in-process but not consumed during the period of production.

Prepared Ingredients

Ingredients that are combined into a subassembly at a step prior to mixing and precooking.

Production Worker

Individual employed by the enterprise as production labor.

USDA, FDA, and Other Requirements

Includes requirements imposed on the manufacture of food products by federal and state agencies, for example, the USDA meat and poultry inspection regulations.

COMBAT RATION ADVANCED MANUFACTURING TECHNOLOGY DEMONSTRATION (CRAMTD)

Technical Report: Informational Architecture for Packaged Food Manufacturing

Technical Working Paper (TWP) 52

Nabil R. Adam, Thomas O. Boucher, Timothy Chamberlin, and John Weber
Department of Industrial Engineering

Rutgers, The State University of New Jersey

April 1992

Sponsored by:
DEFENSE LOGISTICS AGENCY
Cameron Station
Alexandria, VA 22304-6145

Contractor:
Rutgers, The State University of New Jersey
THE CENTER FOR ADVANCED FOOD TECHNOLOGY*
Cook College
N.J. Agricultural Experiment Station
New Brunswick, New Jersey 08903

DR. JACK L. ROSSEN
Program Director, CRAMTD
DR. JOHN F. COBURN
Associate Director, CRAMTD

TEL: 908-932-7985/8307
FAX: 908-932-8690

*A New Jersey Commission on Science and Technology Center

1.0 Introduction

This report addresses the requirements of Task Items 3.6.4 of STP #4, requiring a technical report on the design of an Informational Architecture for the Packaged Food Industry. Phase II of STP #4 required studying the procedures by which coalition companies operated their enterprises in the manufacture of shelf stable food products. Based on these studies the research team abstracted the common features of the coalition companies studied, thus, developing a generic set of operating procedures. This generic model is referred to as a "Functional Architecture". A Functional Architecture is a description of the functions performed in operating the enterprise and the relationship among these functions as given by the information flows and material flows linking them. The results of this study was published as Technical Working Paper (TWP)#37, "Technical Report: Functional Architecture for Packaged Food Manufacture".

Phase III of STP #4 requires identifying the data requirements necessary to support the processes modeled by the functional architecture. The data requirements are used as a basis for designing a logical relational database model using the IDEF1X methodology. This methodology was developed under funding from the US Airforce. It is an entity - attribute - relationship methodology that has evolved from earlier work by Chen (1976) and Nijssen (1979).

In the next section we present an overview of the IDEF1X modeling methodology. This will be followed by a description of the organization of the IDEF1X documentation.

2.0 IDEF1X Methodology

IDEF1X is a semantic data modeling methodology that defines the meaning of data within the context of its interrelationship with other data. A completed IDEF1X diagram is a static structure that defines information groupings and relationships among these groupings. IDEF1X uses the entity-relationship approach to semantic data modeling. That is, an IDEF1X model has three basic components: Entities, attributes, and relationships.

An Entity is an element (Part) of the system that is of relevance to our study. It can be something abstract such as "Contract Number" or something tangible, e.g. " vendor lot", which refers to the lots of raw ingredients or materials supplied by vendors. Entities can be classified into different entity types, e.g. equipment, products, and purchase orders. A collection of entities of the **same type** make up an **entity set** whose members are referred to as instances of that entity set. For example, a products entity set has several instances, each representing a given product.

In IDEF1X, an entity set is represented by a Box. Figure 1 shows the basic diagrammatic structure of IDEF1X.

An entity set has properties (characteristics) called **Attributes,** e.g. Name and Address of the "Vendor" entity set. All entities in a given entity set have the same attributes but, clearly, the values may differ from one instance to another.

An attribute of an entity set, for which each instance must have a unique value is called a "key attribute" for that entity set. For example, each instance of the vendor lot entity class

has a unique material lot number.

In the IDEF1X diagram, entity attributes are listed within the box representing that entity. The primary key(s) of a given entity are separated from the rest of the attributes by a line that goes across the box.

Relationships may exist between entities. For example, the "vendor lot" entity set is related to the quality "tests" entity set in the sense that each vendor lot is inspected according to one or more quality tests. At the same time a given quality test is used to inspect more than one vendor lot. The result of applying a specific quality test (whose primary key is TEST ID) on a specific vendor lot (whose primary key is the MATERIAL LOT No) is represented by a quality report. There are several instances of quality reports, each corresponding to a given combination of a vendor lot and a quality test. A key attribute that provides the linkage between entities is called a "foreign key" (FK). For example, the foreign key that relates vendor lot with quality report is "material lot no".

A relationship has a Cardinality, which specifies the number of instances of an entity with which a given entity is associated through that relationship. There are three possible cardinalities: one to one (1:1), one to many (1:N), and many to many (M:N). For example, "Vendor" and "Vendor Lot" have a one to many relationship, i.e. a vendor supplies several vendor lots and a given vendor lot is supplied by only one vendor.

IDEF1X allows the cardinality of a relationship to be indicated by the arc joining the entities. For example, a

specific material lot no. may be inspected using one, two, or more quality tests and, therefore, has one, two, or more quality reports associated with it. A solid arc with a dot, as shown in Figure 1, denotes zero, one, or many. By attaching a specific number to a dot, the cardinality can be made specific. Where there are no dots, the relationship is one to one. An entity that relates to zero, one, or many instances of another entity is a "parent" entity to that "child entity". "Vendor lot" is a parent entity to the child entity "quality report".

Figure 1 can be simply described using an english syntax: "Each vendor lot is inspected by zero, one, or many quality tests and the result of each test performed on a given vendor lot is recorded in a given quality report".

An IDEF1X model can be easily read by business professionals without any special computer training. As in the case of IDEFO, the graphical representation allows CIM system engineers, management, and those who work within the manufacturing enterprise to communicate ideas with each other as the design process proceeds.

Given this brief introduction to the basic concepts of the IDEF1X methodology, the reader should be able to understand the diagrams included in this document. For a more detailed discussion related to IDEF1X, the reader is referred to reference (3).

3.0 Organization of IDEF1X Documentation

This document is organized into five sections entitled:

1. Manage Contracts, Orders and Bidding Process

2. Plan for Manufacture,

3. Manufacture Product,

4. Control Manufactured Products, and

5. Summary.

In the first four sections, the IDEF1X diagrams are based on the IDEFO functional model published as Technical Working Paper (TWP) 37. For relevant functions in which data is being used or created, the IDEF1X models in this document show the Entities that are being accessed by the decision maker. The particular function is identified with the identifying number and name that is used in the IDEFO Documentation, (TWP) 37. Therefore, the reader can examine the IDEF1X model for a particular function and then go to the IDEFO Document (TWP) 37 to obtain more details concerning what the function does. Thus, both models are traceable to each other.

It should be noted that, when an entity is used in a particular function, the entire entity is shown. In fact, in most cases only a subset of the attributes of the entity is being used. That particular subset is not explicitly identified, although it will be fairly obvious to the reader as to what attributes are relevant.

Section 5, the Summary IDEF1X diagram, is a foldout of the complete IDEF1X structure. It shows the relationship among all entities of the model. It also includes a glossary of definitions for all attributes used in the model.

It should be noted that an IDEF1X model is a logical database design. Thus, each of the entities depicted by a box in the IDEF1X diagram is represented by a table in the database

implementation. Besides the functions explicitly modeled by IDEF0, there are other reports and information that can be generated from the IDEF1X model, such as quality reports and ingredient yields.

The IDEF1X model is the basis for a preliminary database design. This preliminary design will be reported in a future Technical Working paper.

References

1. Boucher, T.O., M.A. Jafari, S. Kim, and J. McPhail, 1991 "Technical Report: Functional Architecture for Packaged Food Manufacturing", TWP 37.

2 Chen, P., 1976, "The Entity-Relationship Model Toward a Unified View of Data", ACM Transactions on Database Systems, 1, 9-36.

3. IDEF1X, 1985, IDEF1X Information Modeling Manual, ICAM Program Office, Wright Patterson Air Force Base.

4. Nijssen, G.M., 1979, "Modeling in Database Management Systems", Proceedings of the Euro IFIP Conference.

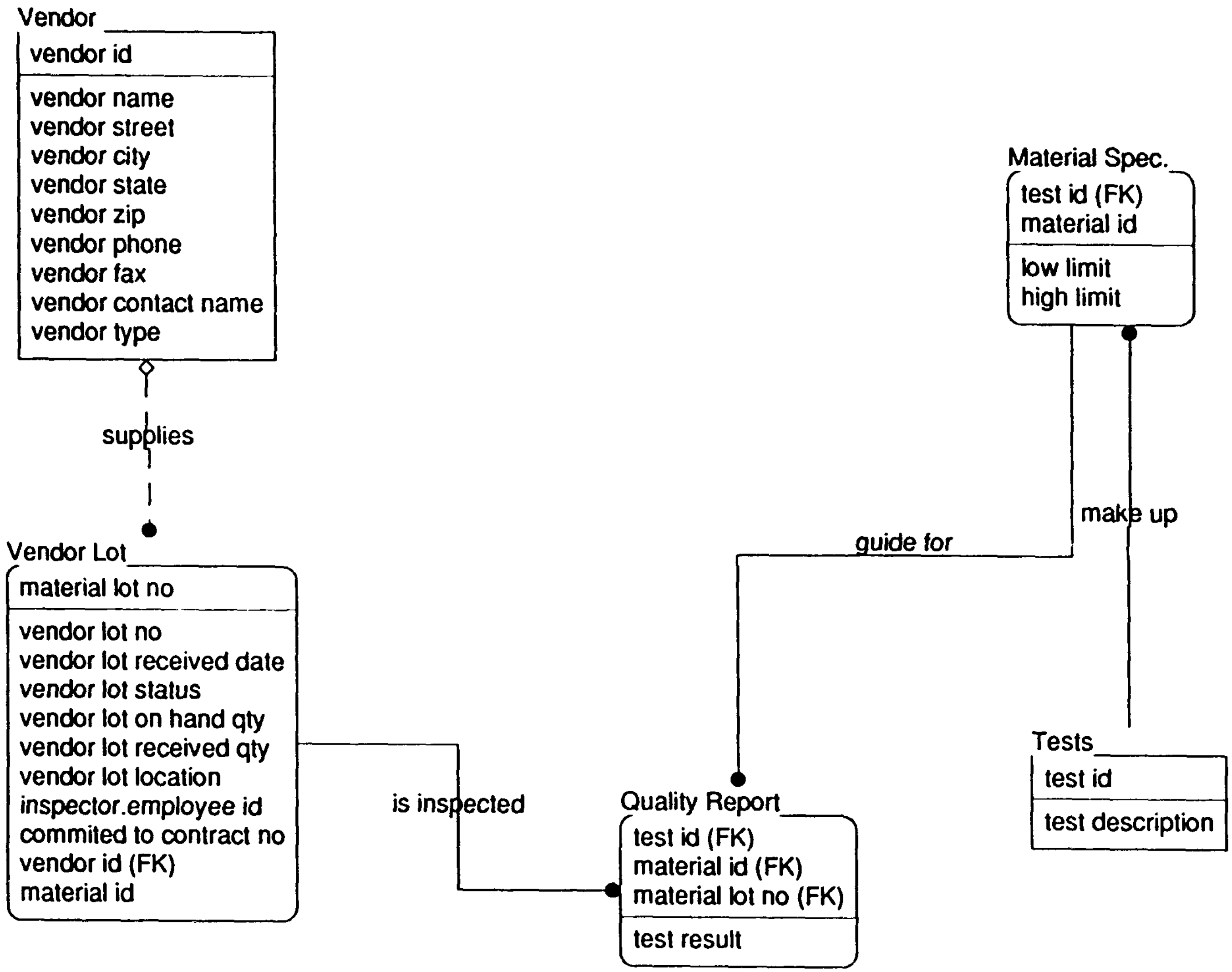
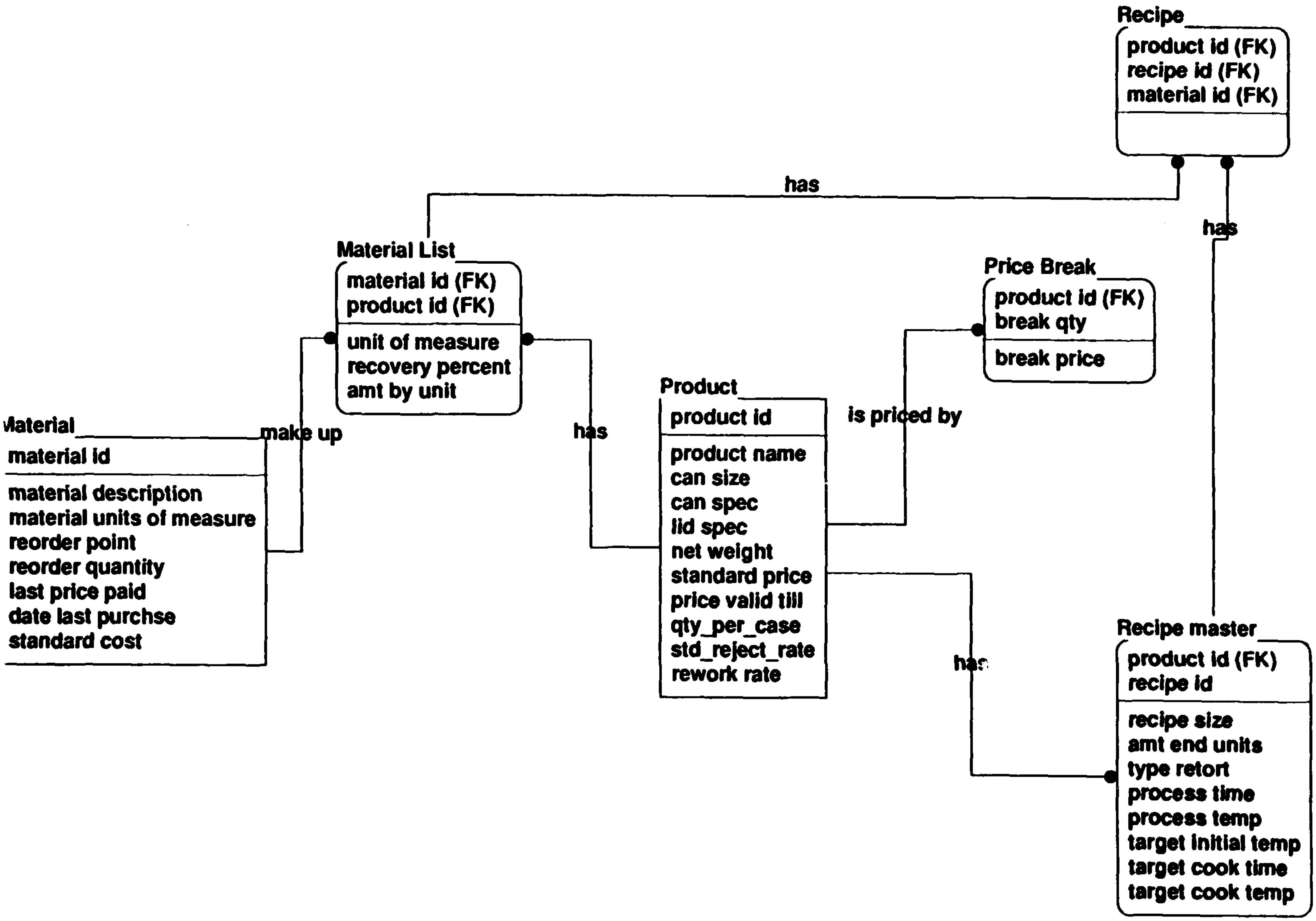


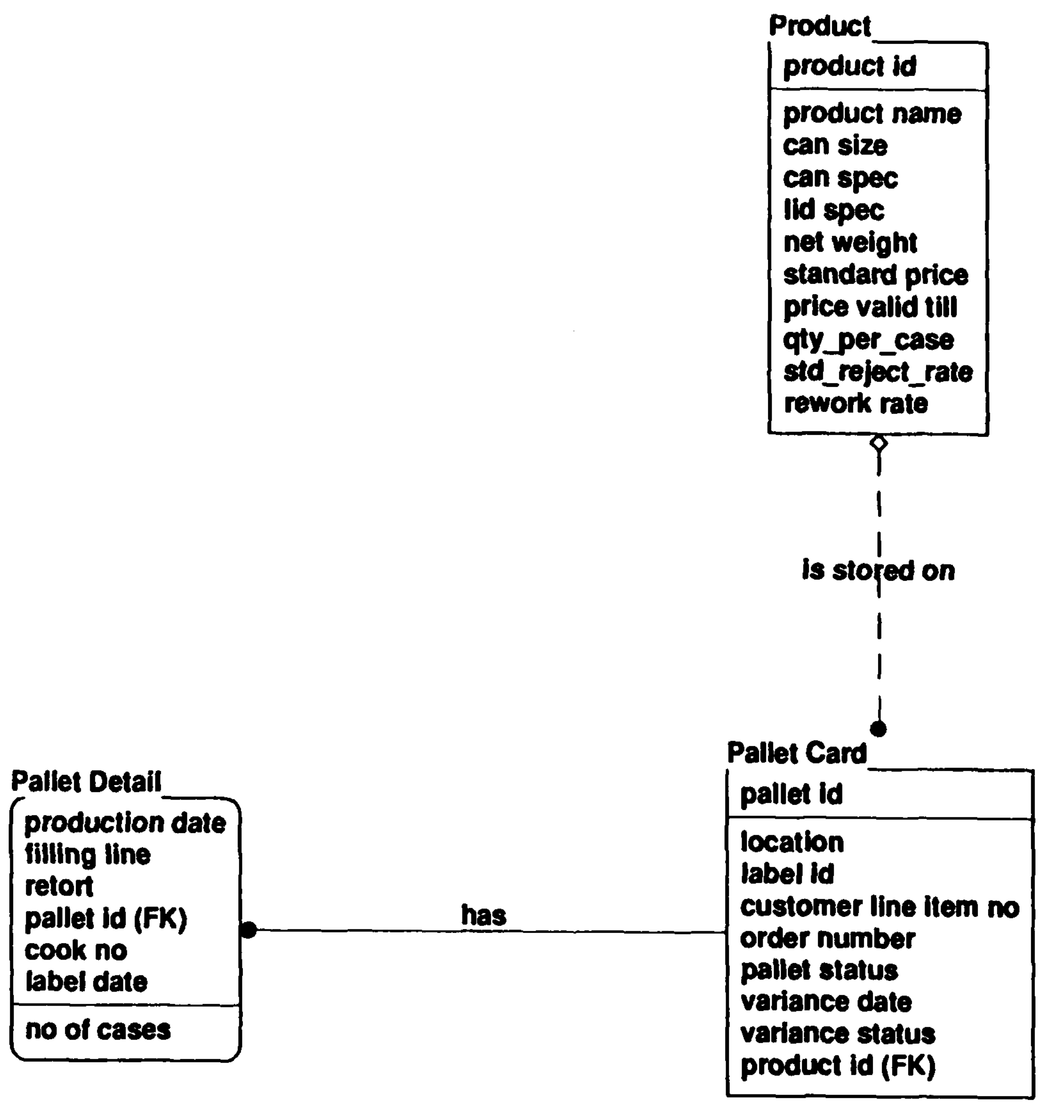
Figure 1: Elements of IDEF1X Modeling

Section I

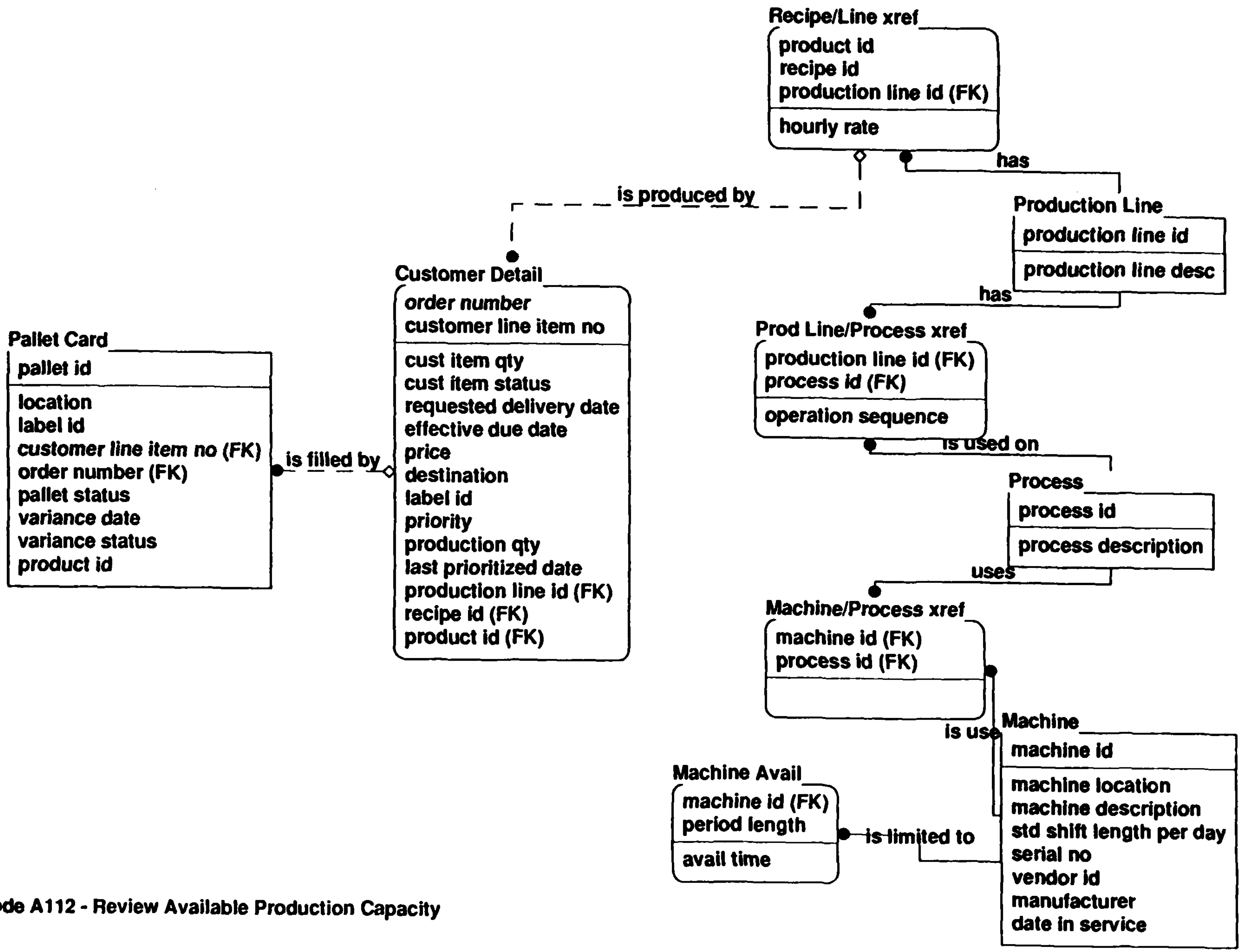
Manage Contracts, Orders, and Bidding Process



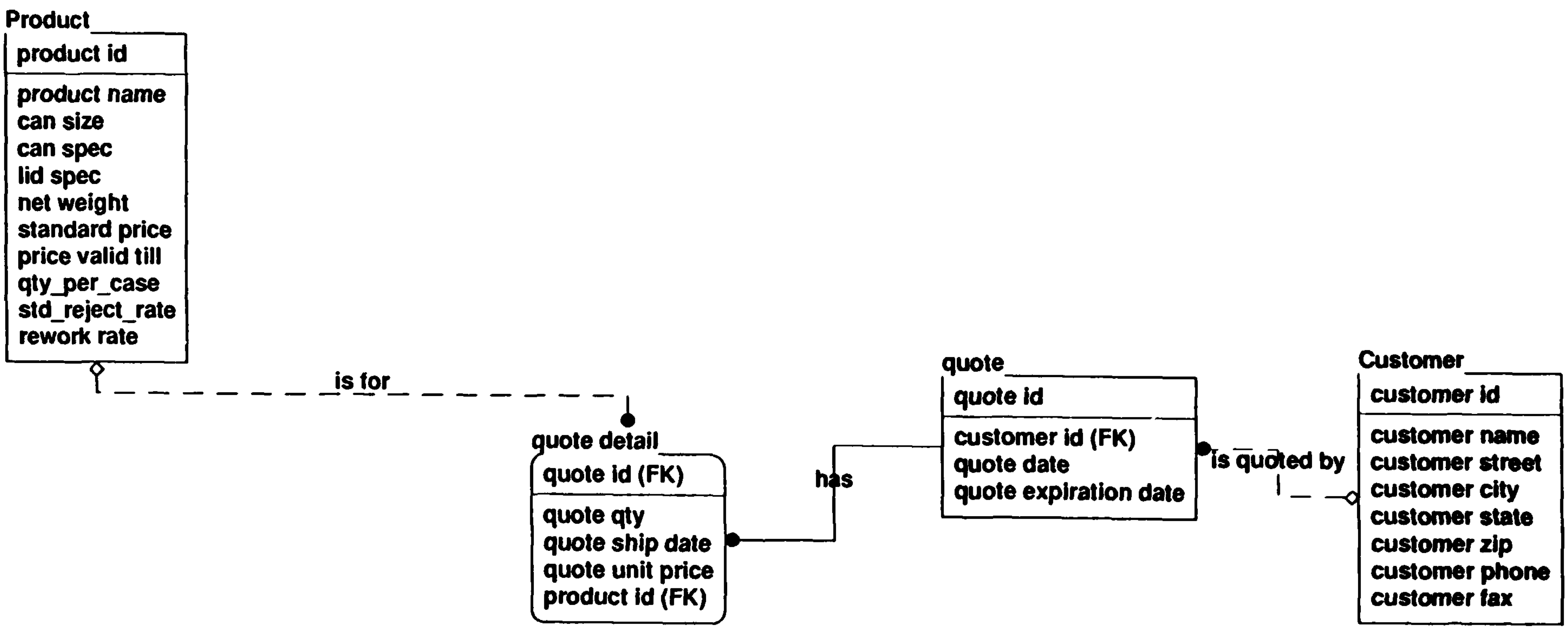
ode A111 - Review Current Pricing And Material Costs



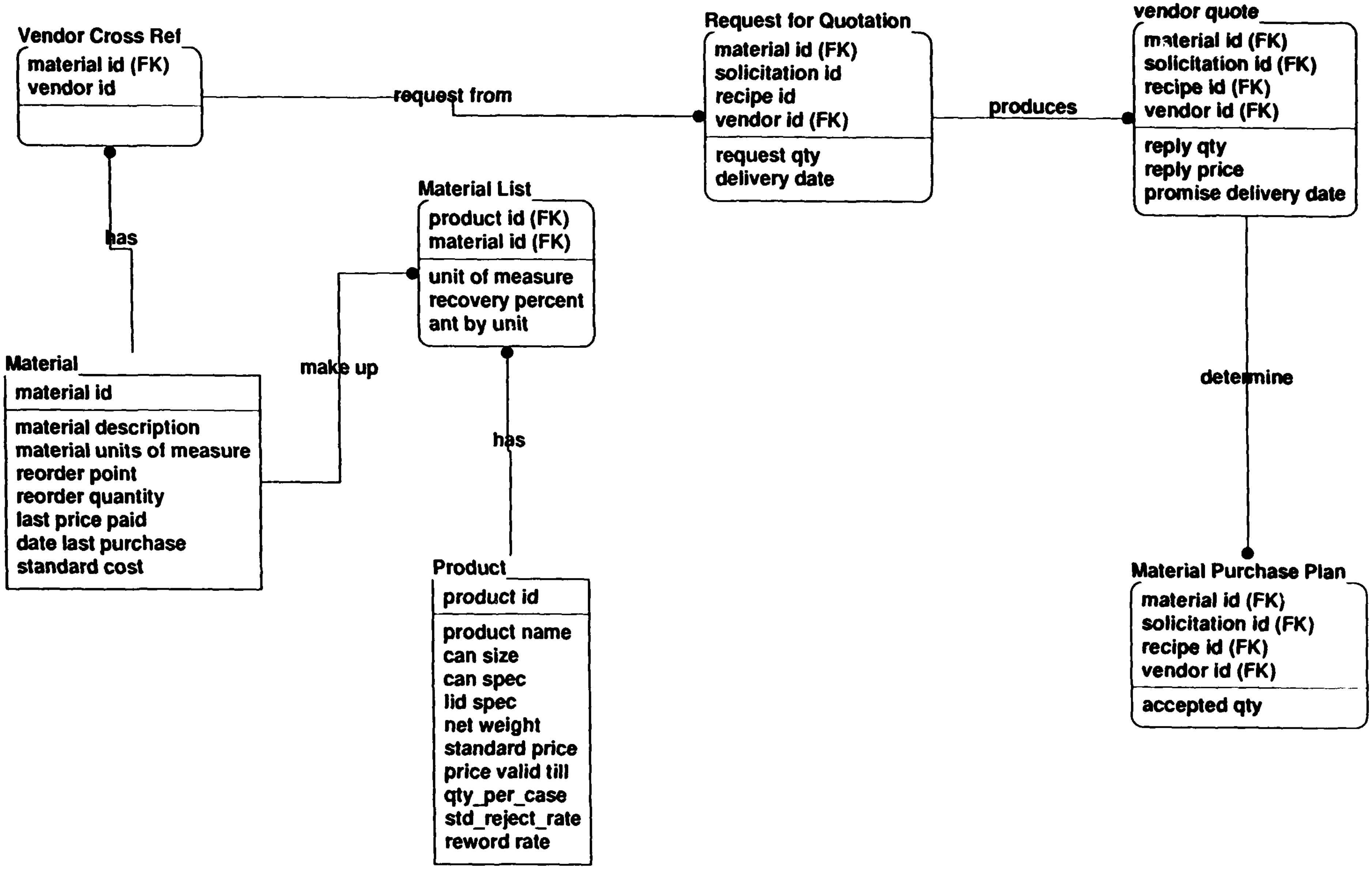
ode A112 - Review F G Inventory



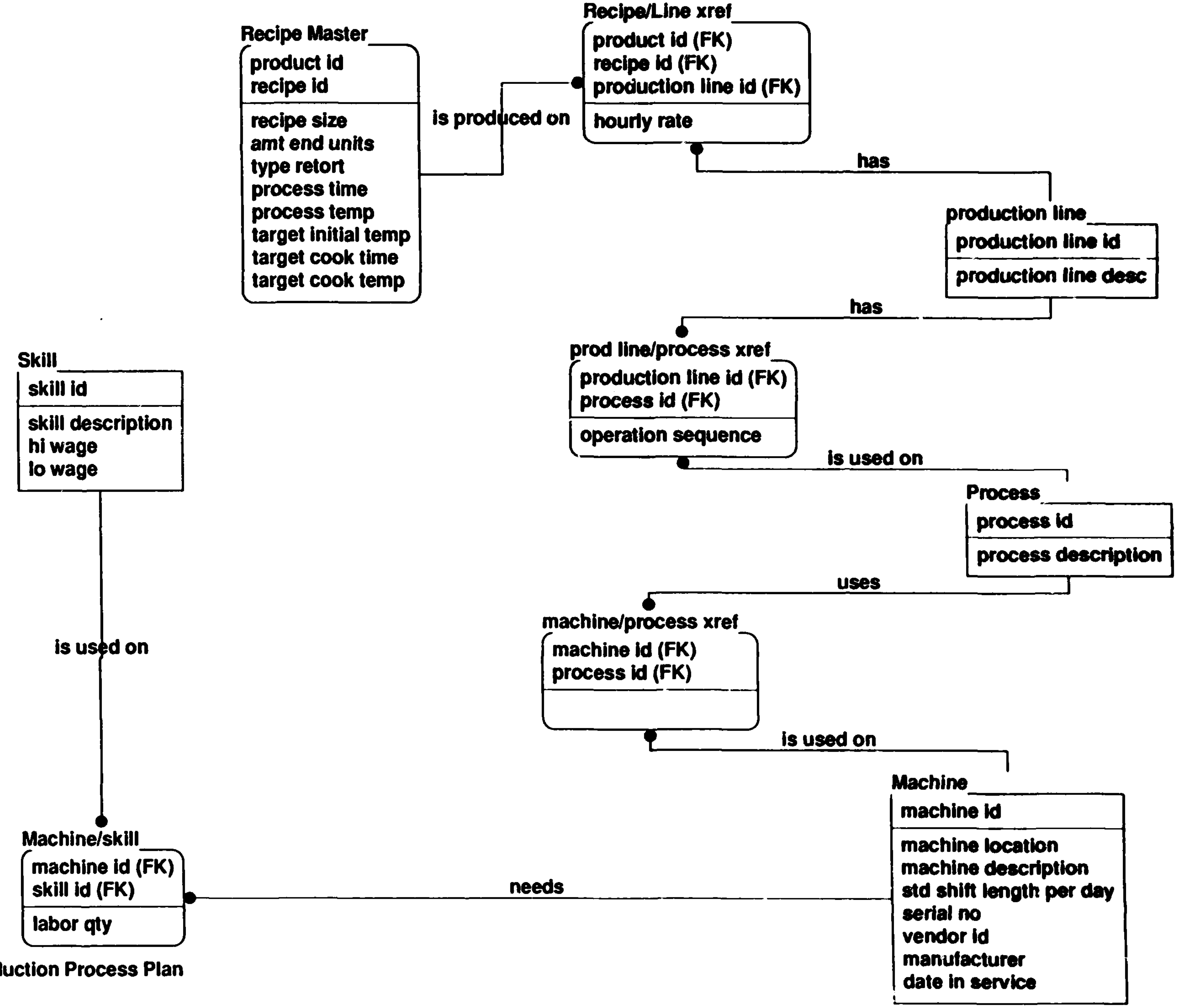
Node A112 - Review Available Production Capacity



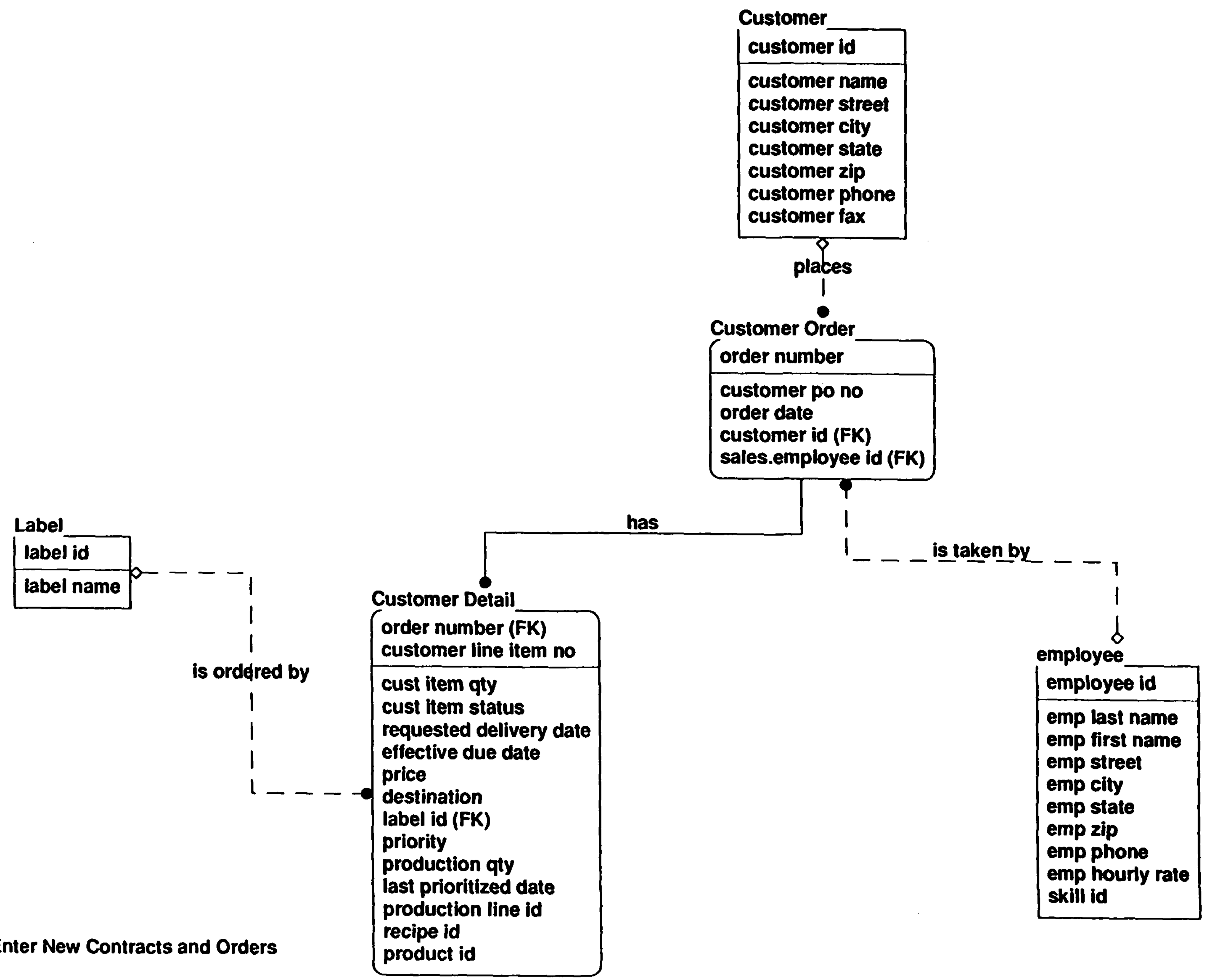
Node A113 - Issue Quote



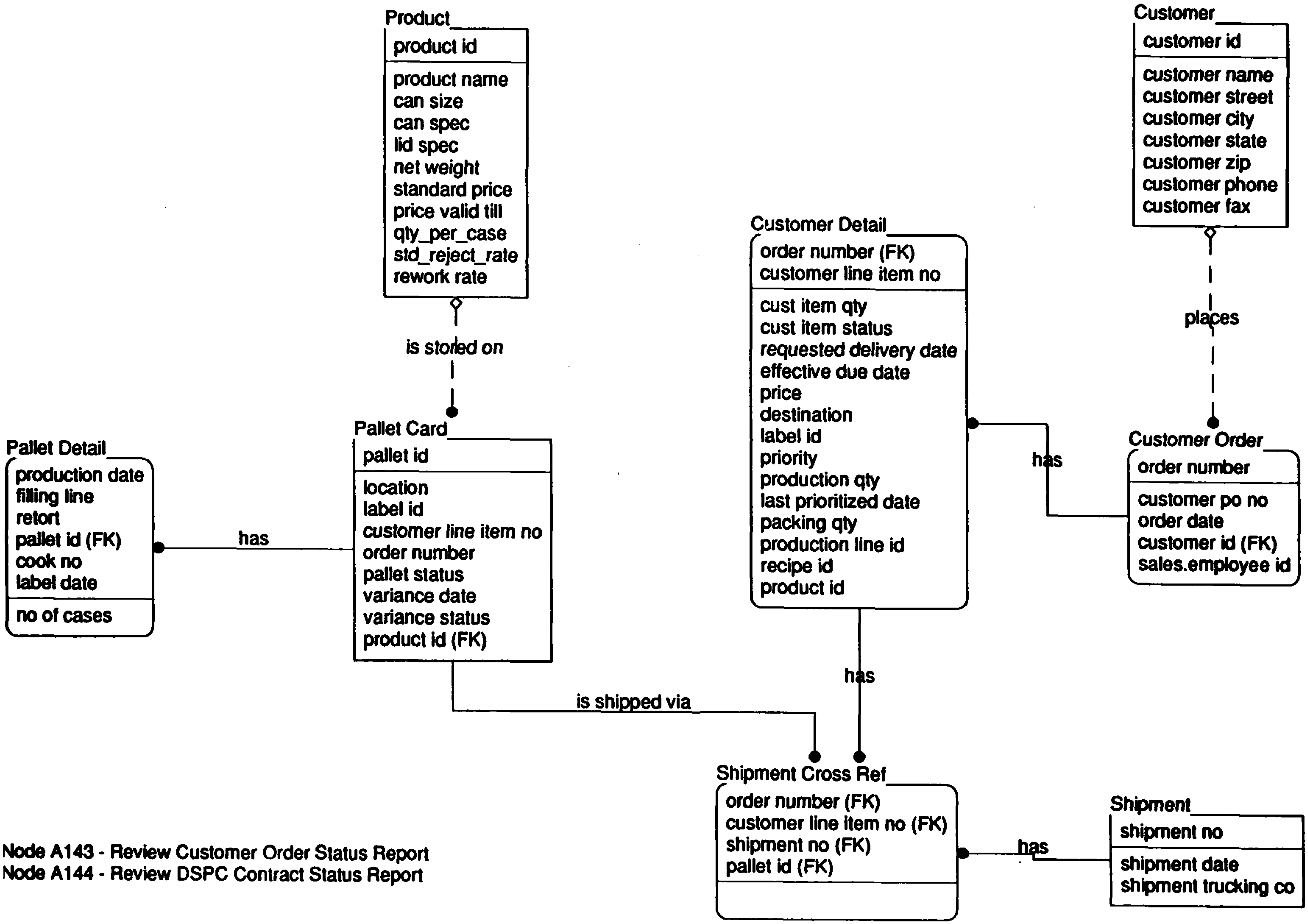
de A1321 - Estimate Material Cost



Node A1322 - Develop Production Process Plan



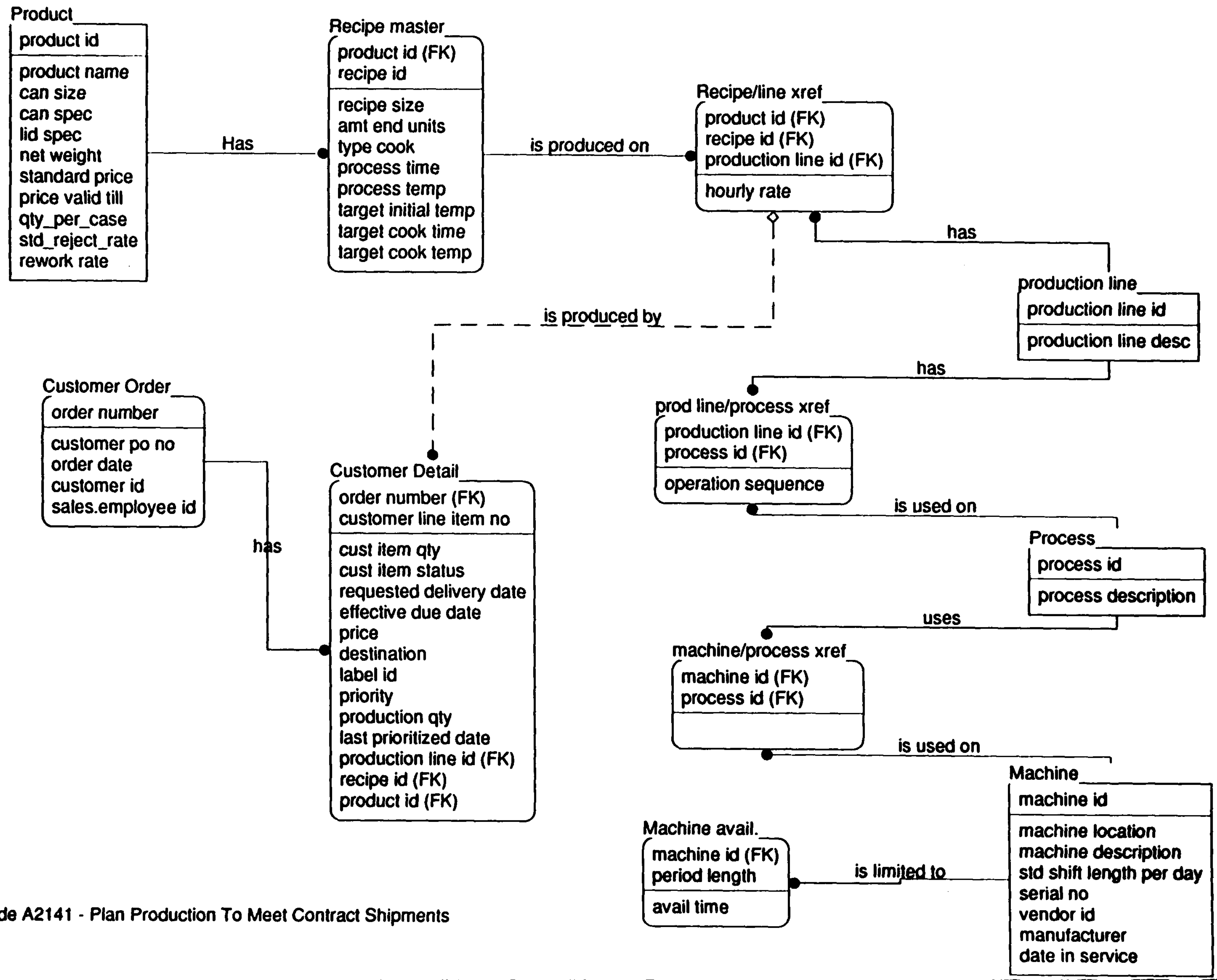
Node A141 - Enter New Contracts and Orders



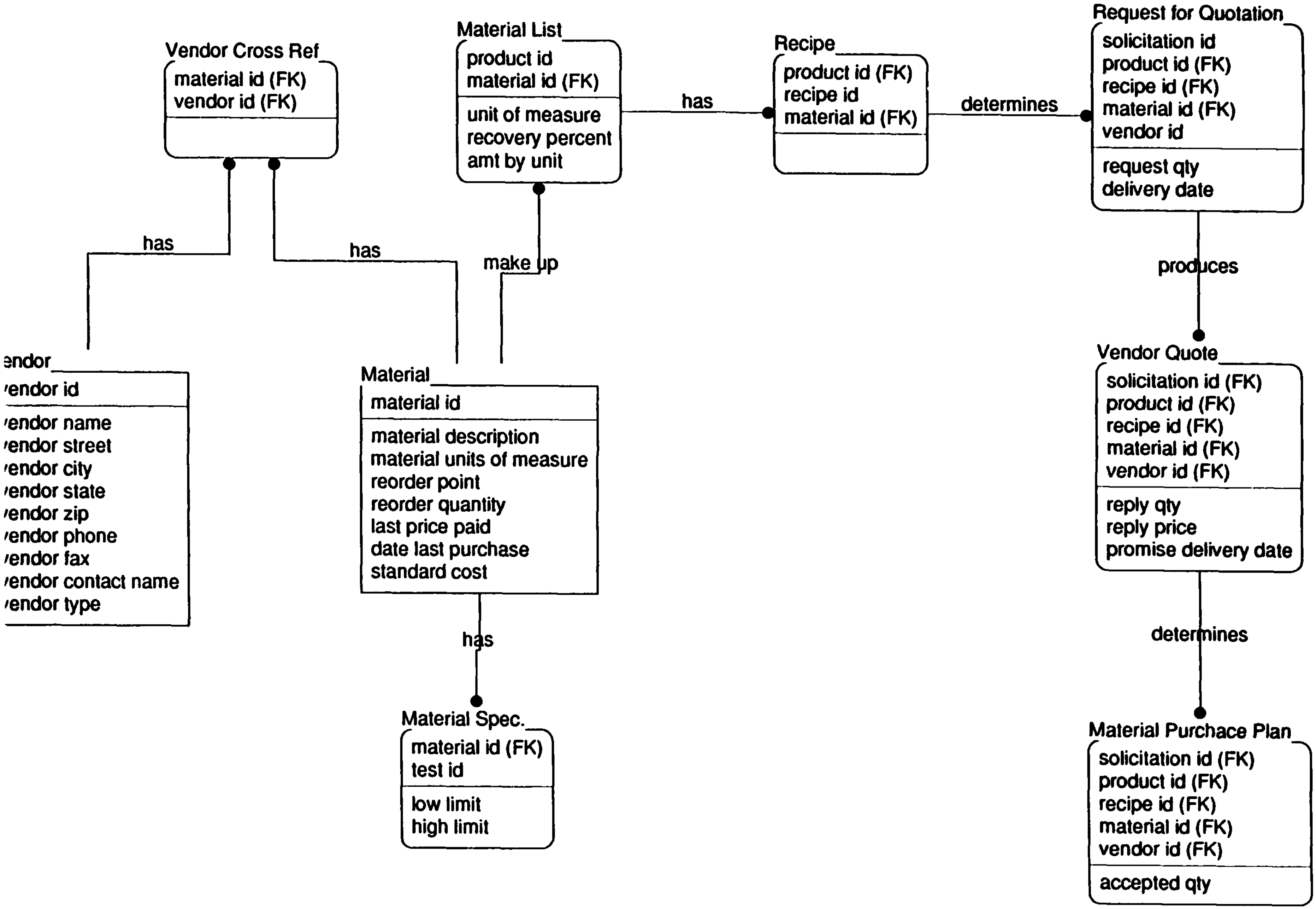
Node A143 - Review Customer Order Status Report
Node A144 - Review DSPC Contract Status Report

Section II

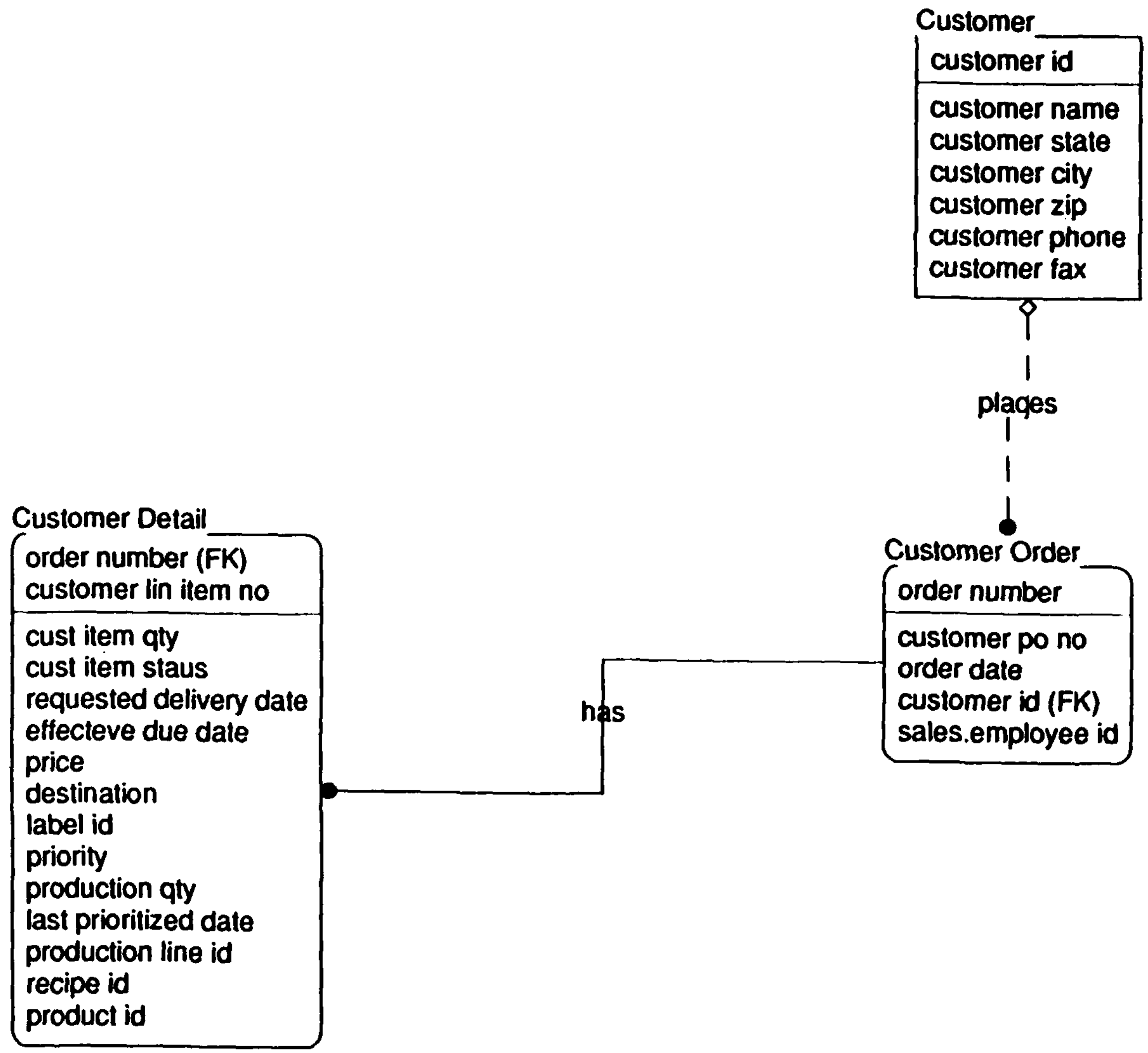
Plan for Manufacture



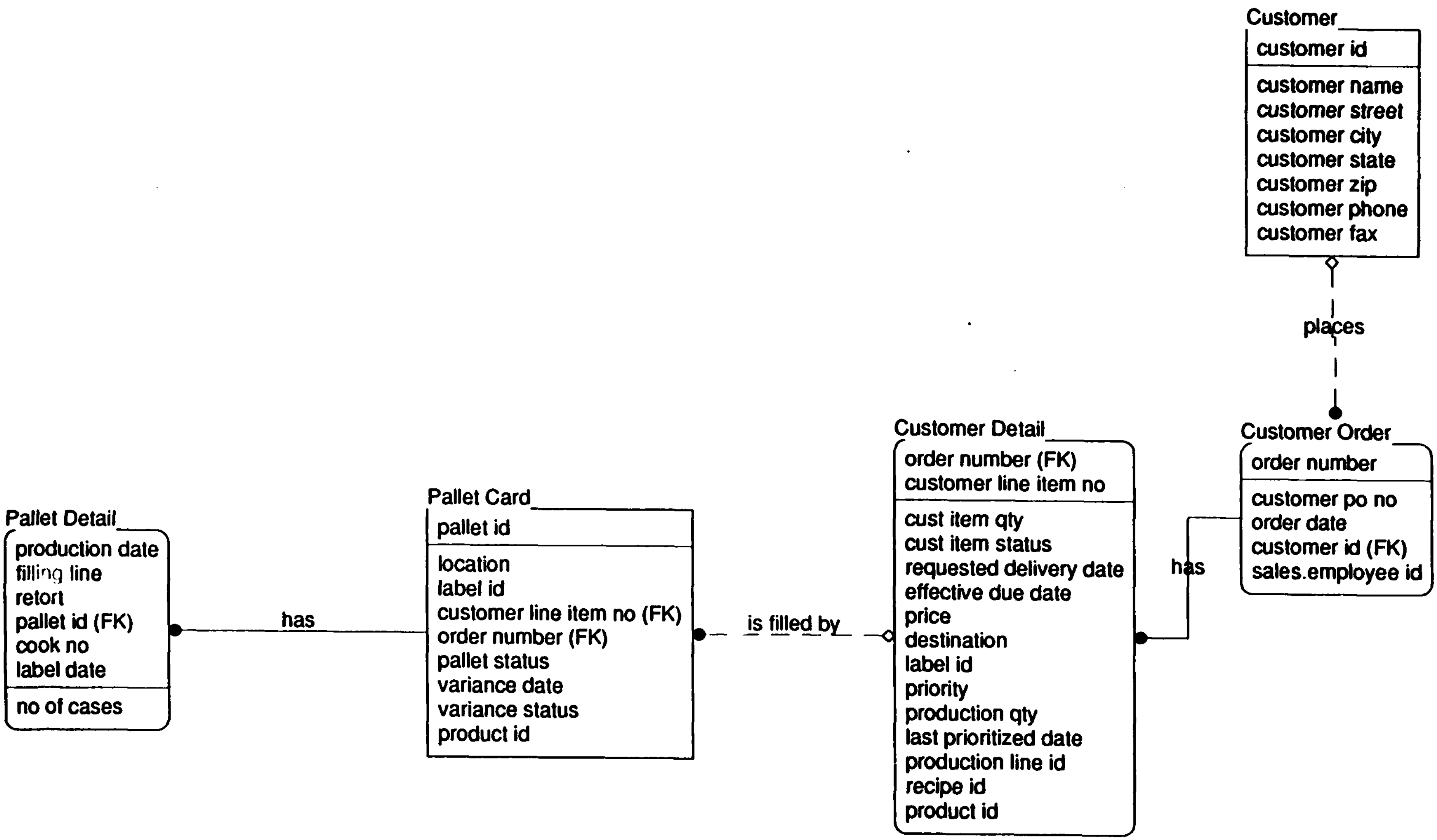
Node A2141 - Plan Production To Meet Contract Shipments



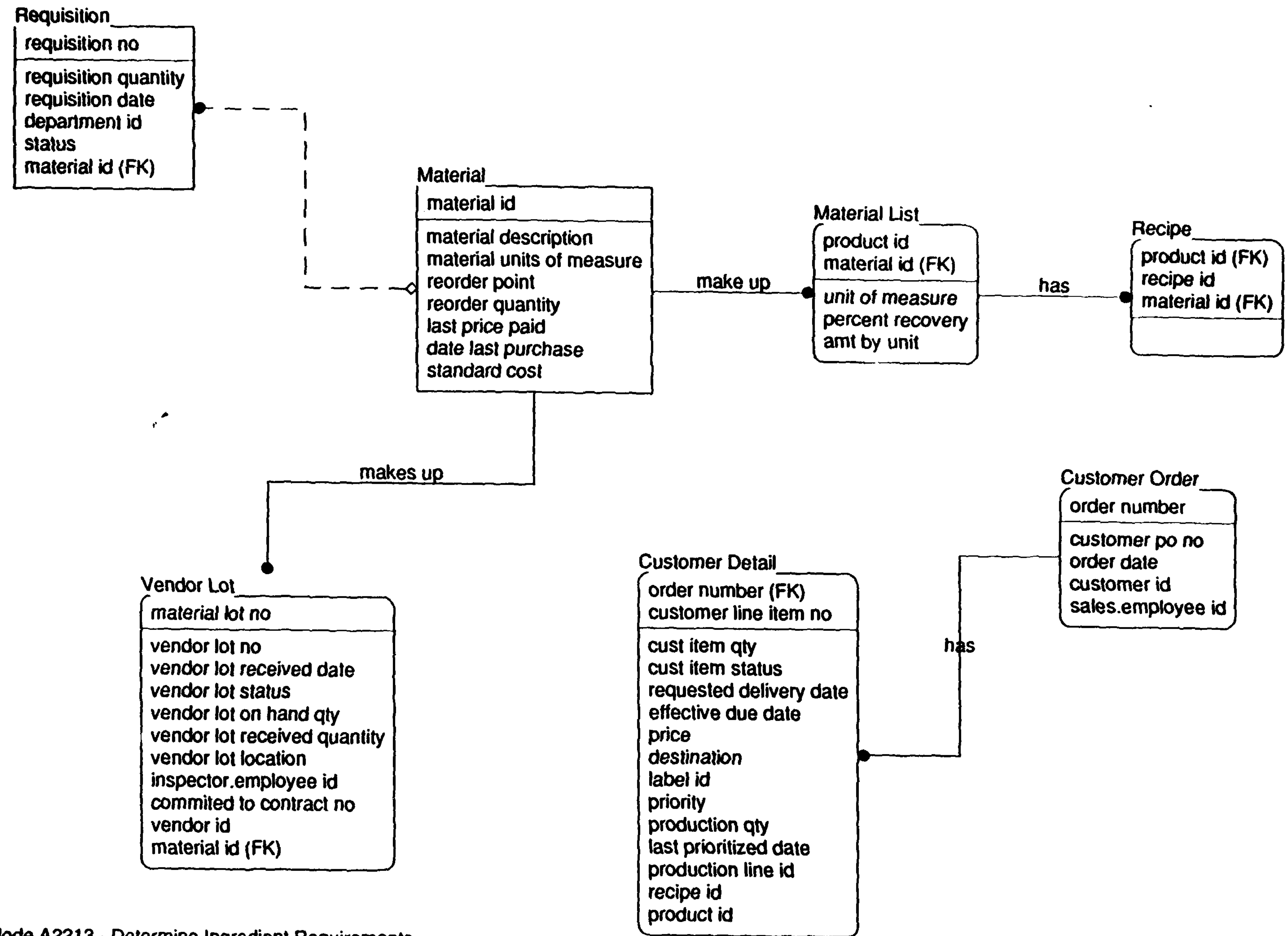
Node A2142 - Develop Materials Plan



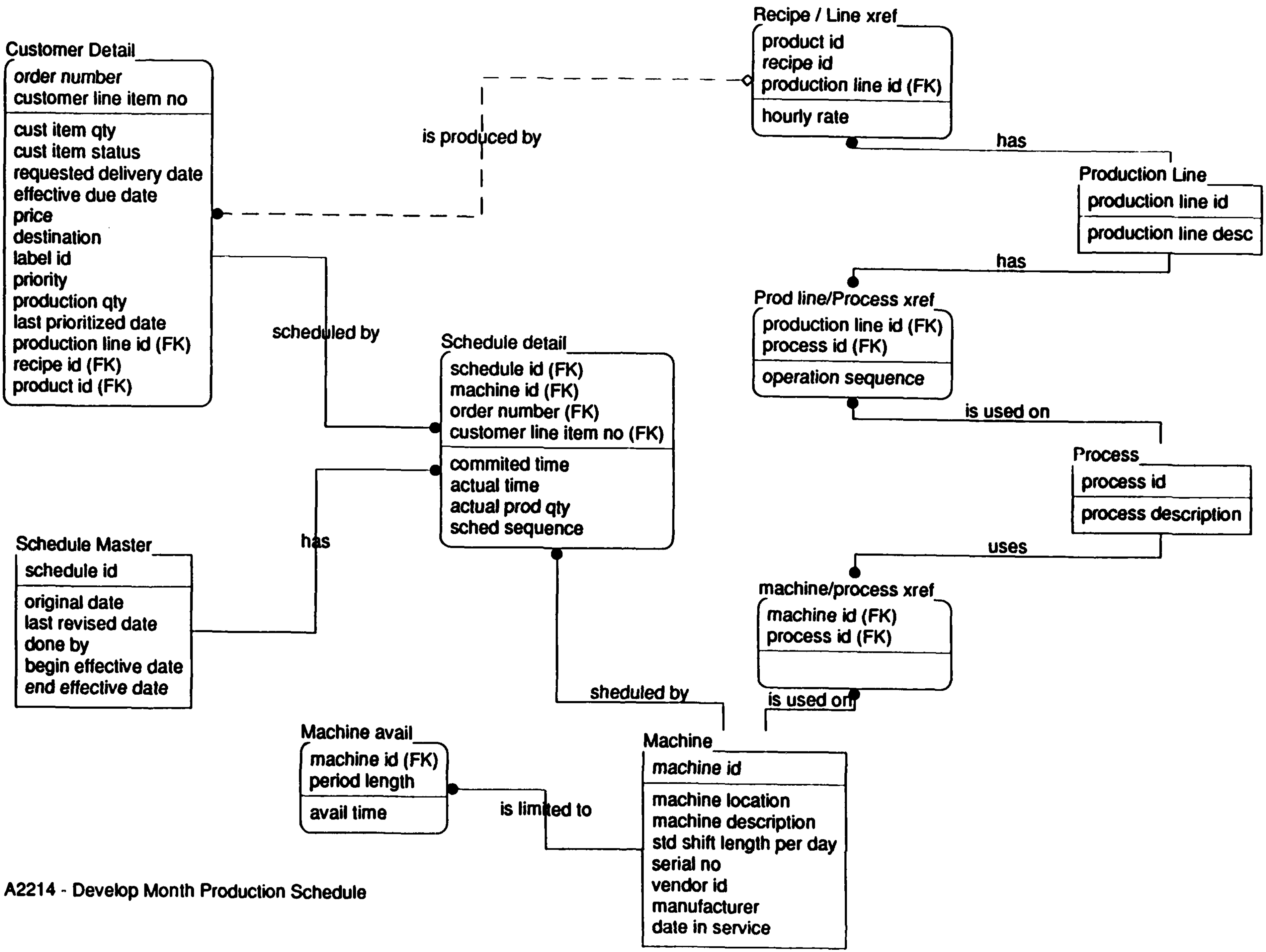
Node A2211 - Prioritize Orders for Production



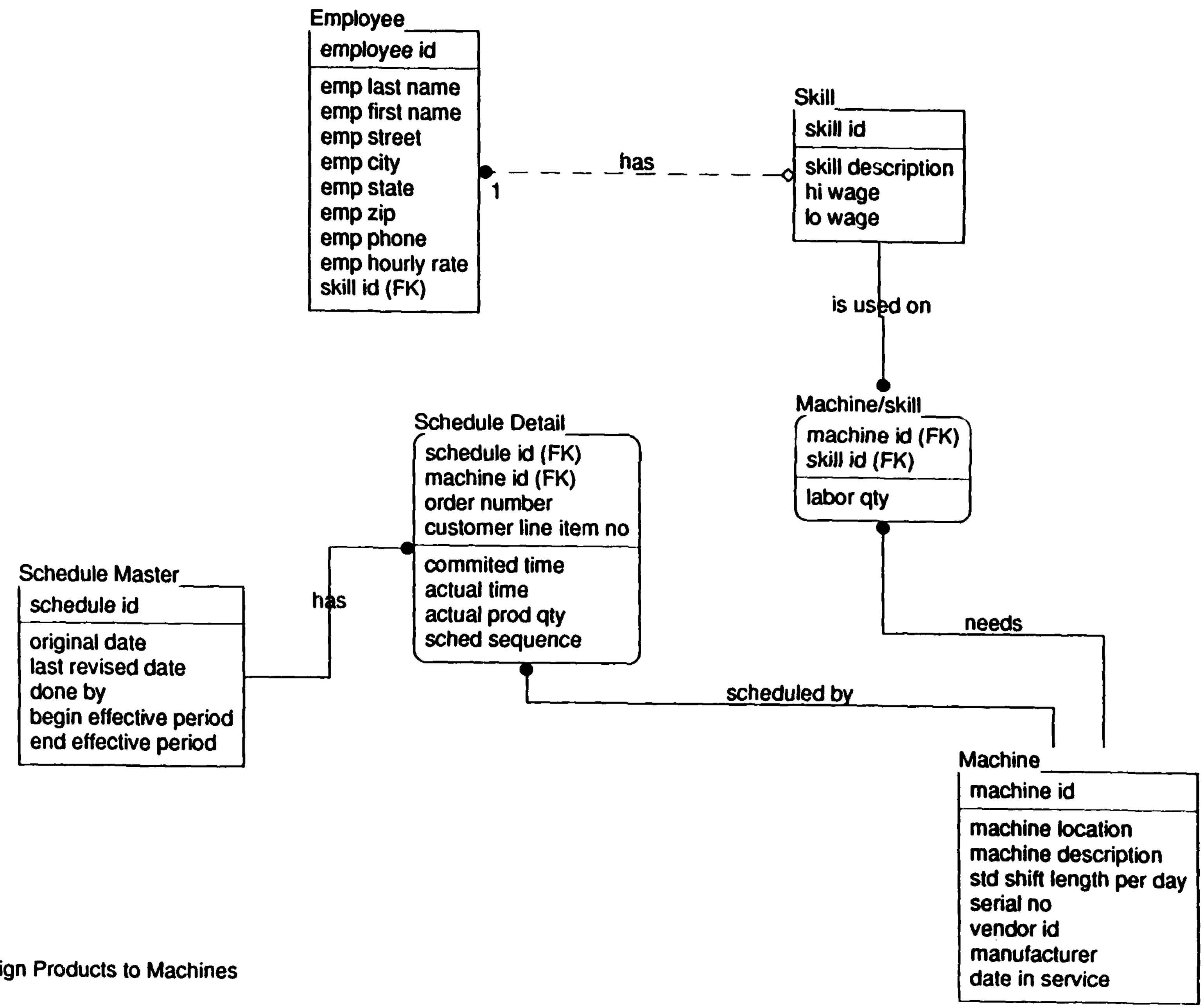
Node A2212 - Determine Net Requirements



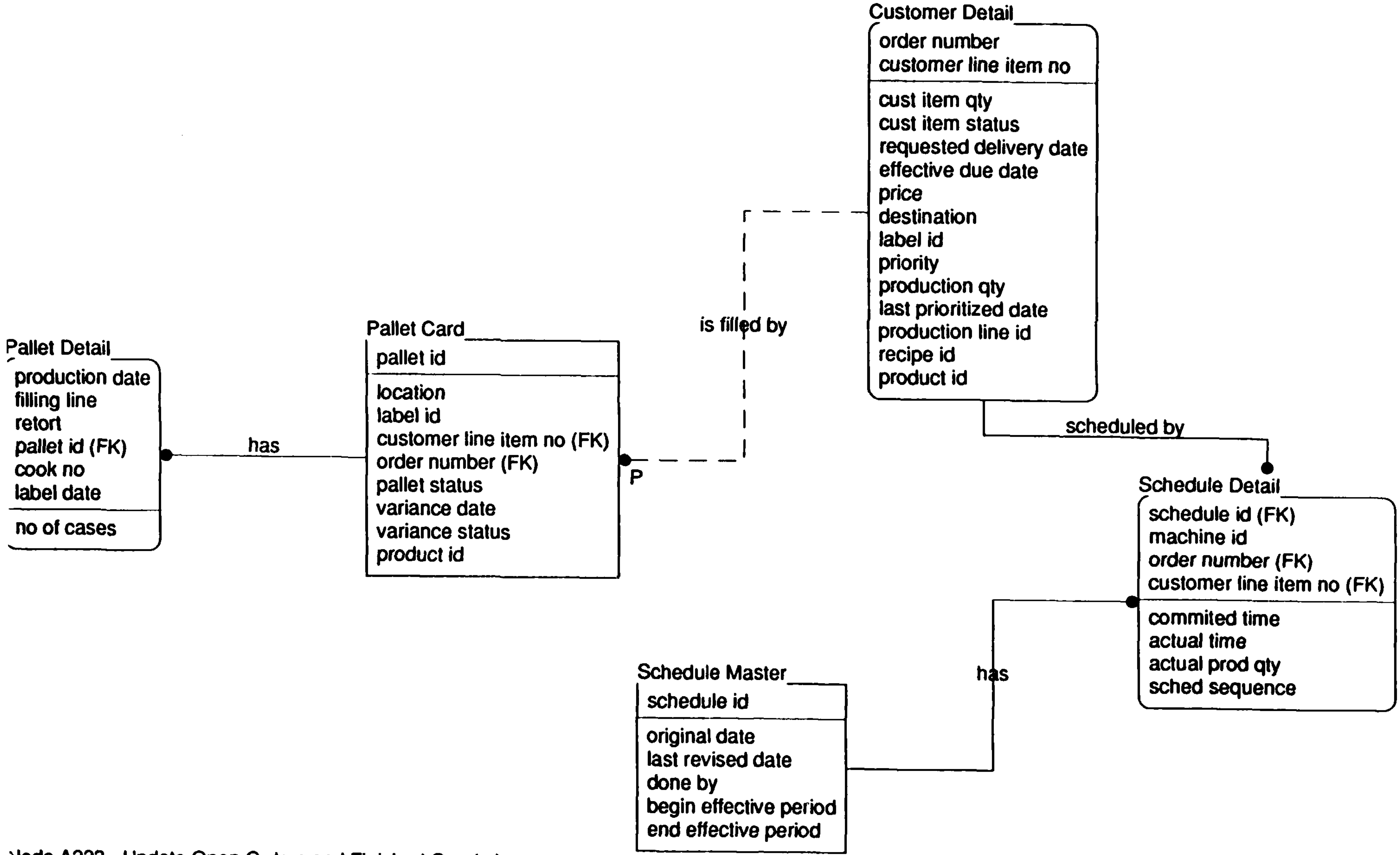
Node A2213 - Determine Ingredient Requirements



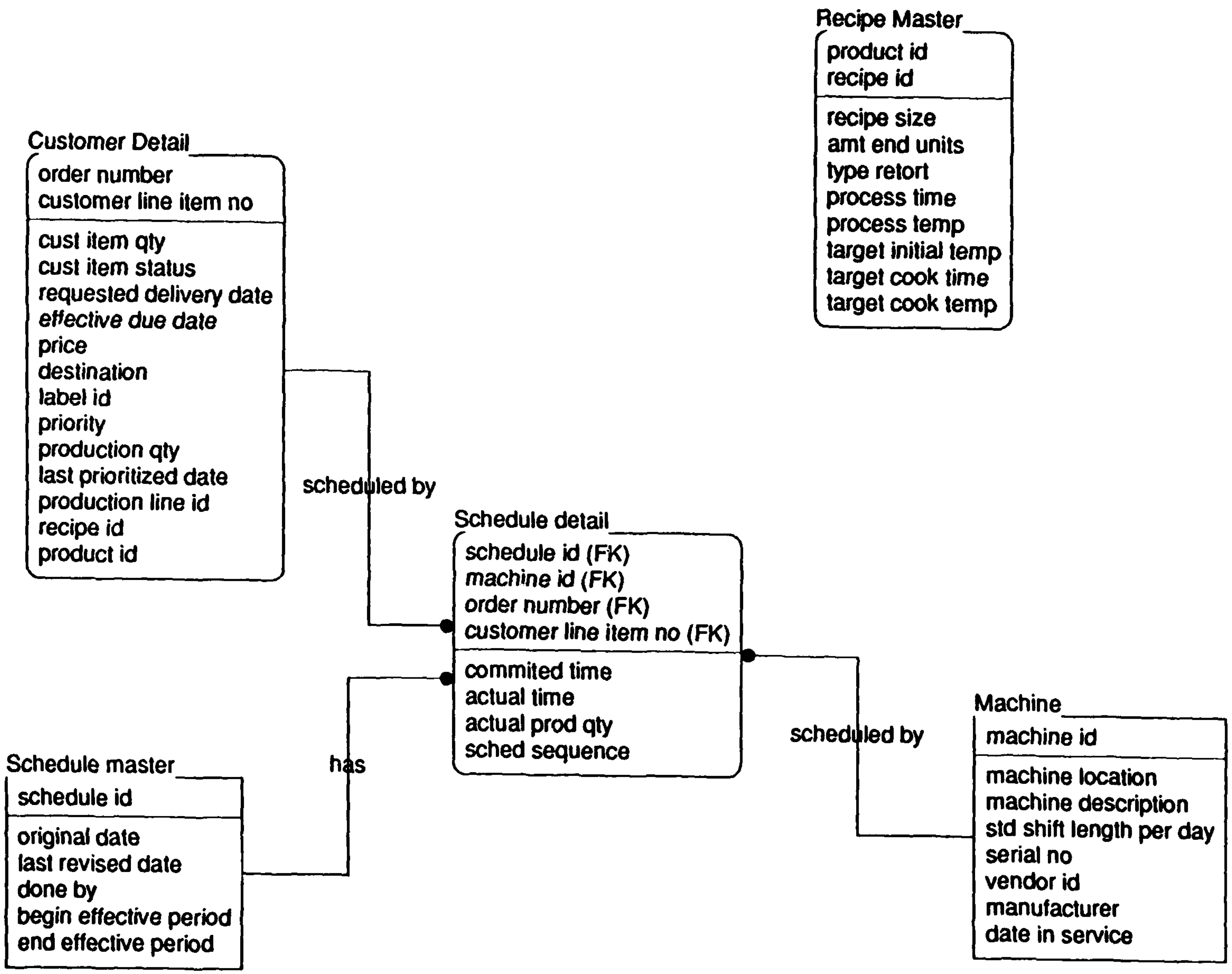
Node A2214 - Develop Month Production Schedule



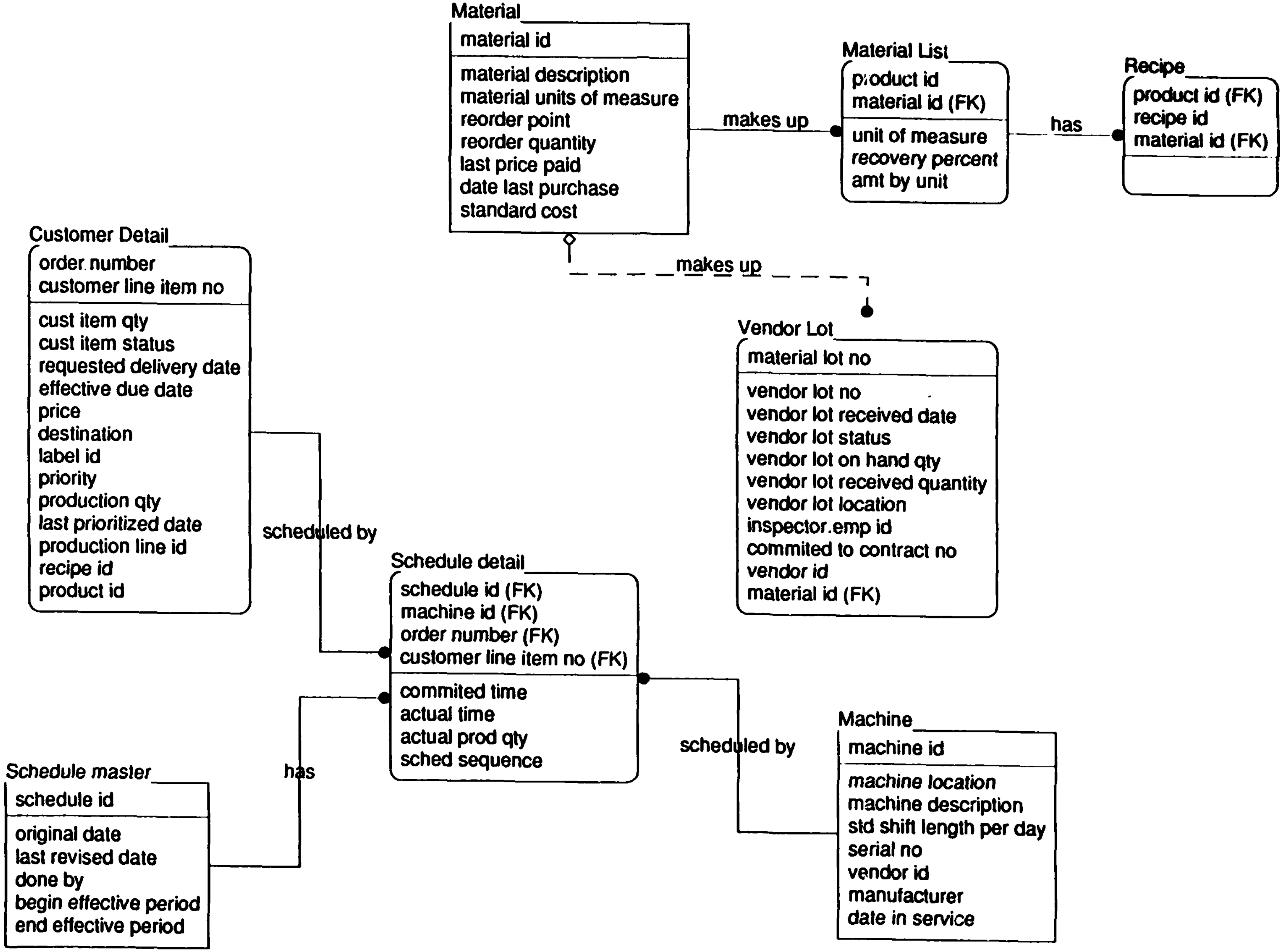
lode A2221 - Assign Products to Machines



Node A223 - Update Open Orders and Finished Goods Inventory



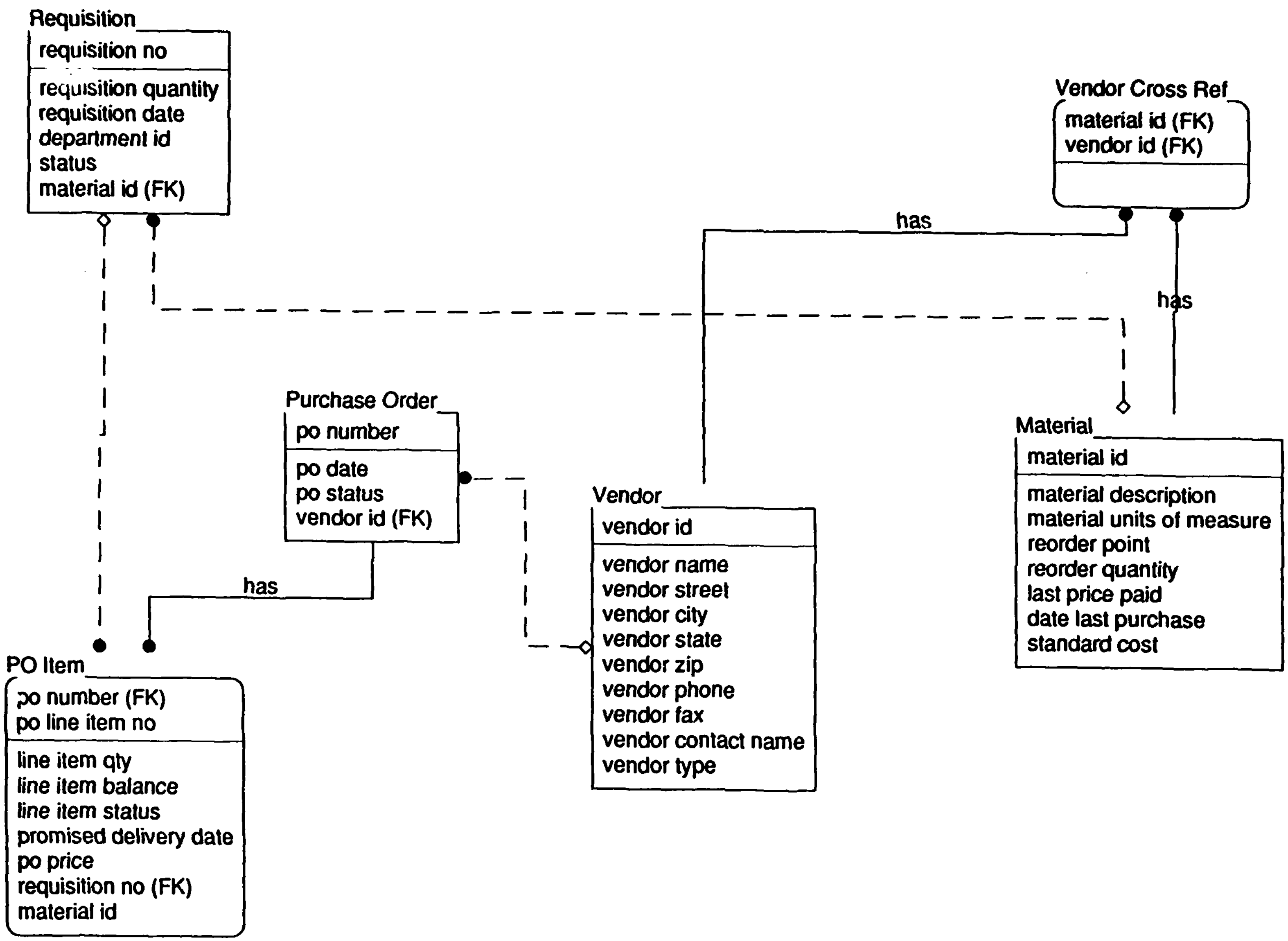
Node A2223 - Develop Retort Schedules



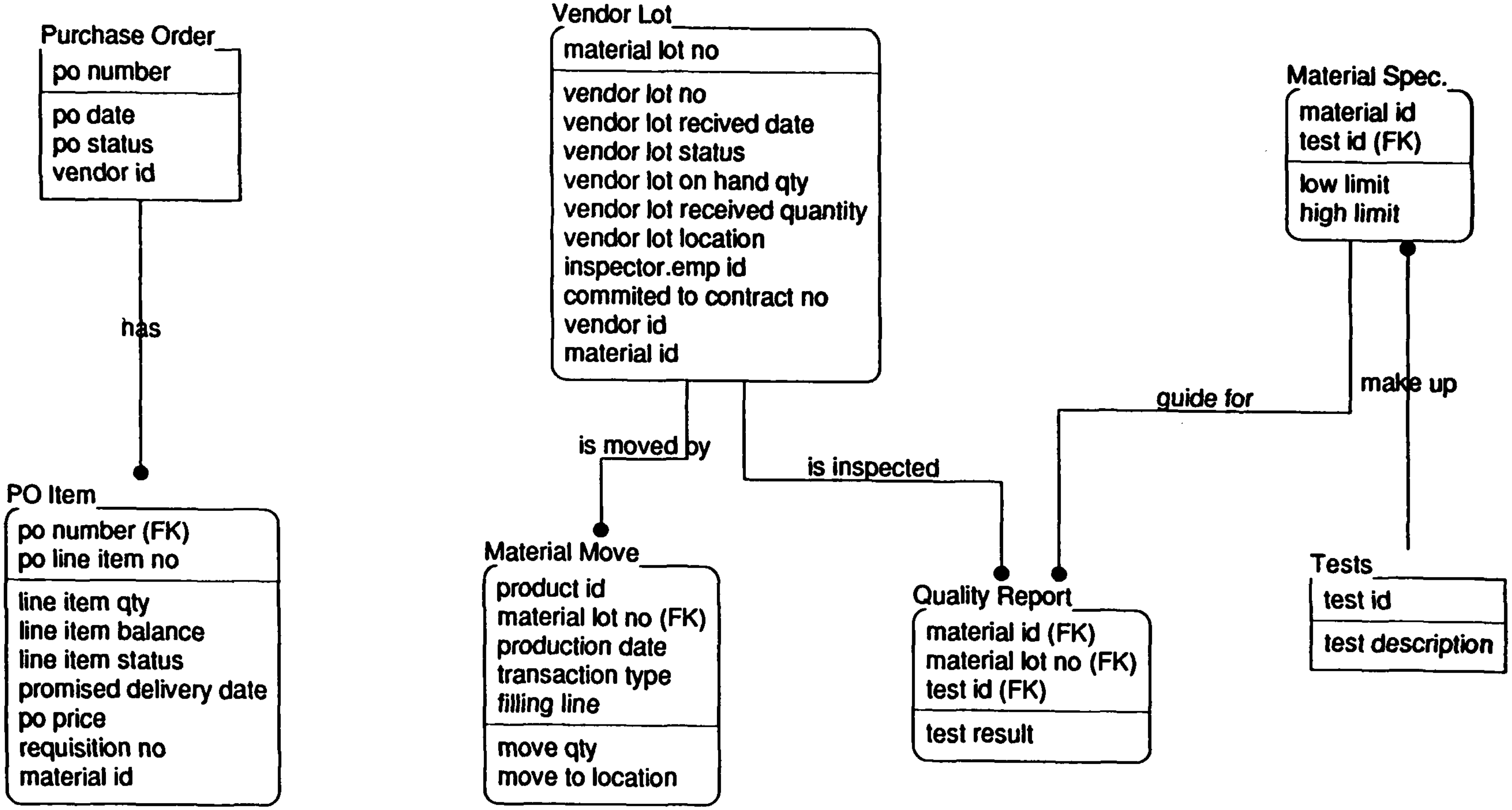
ode A2224 - Develop Material move Schedule

Section III

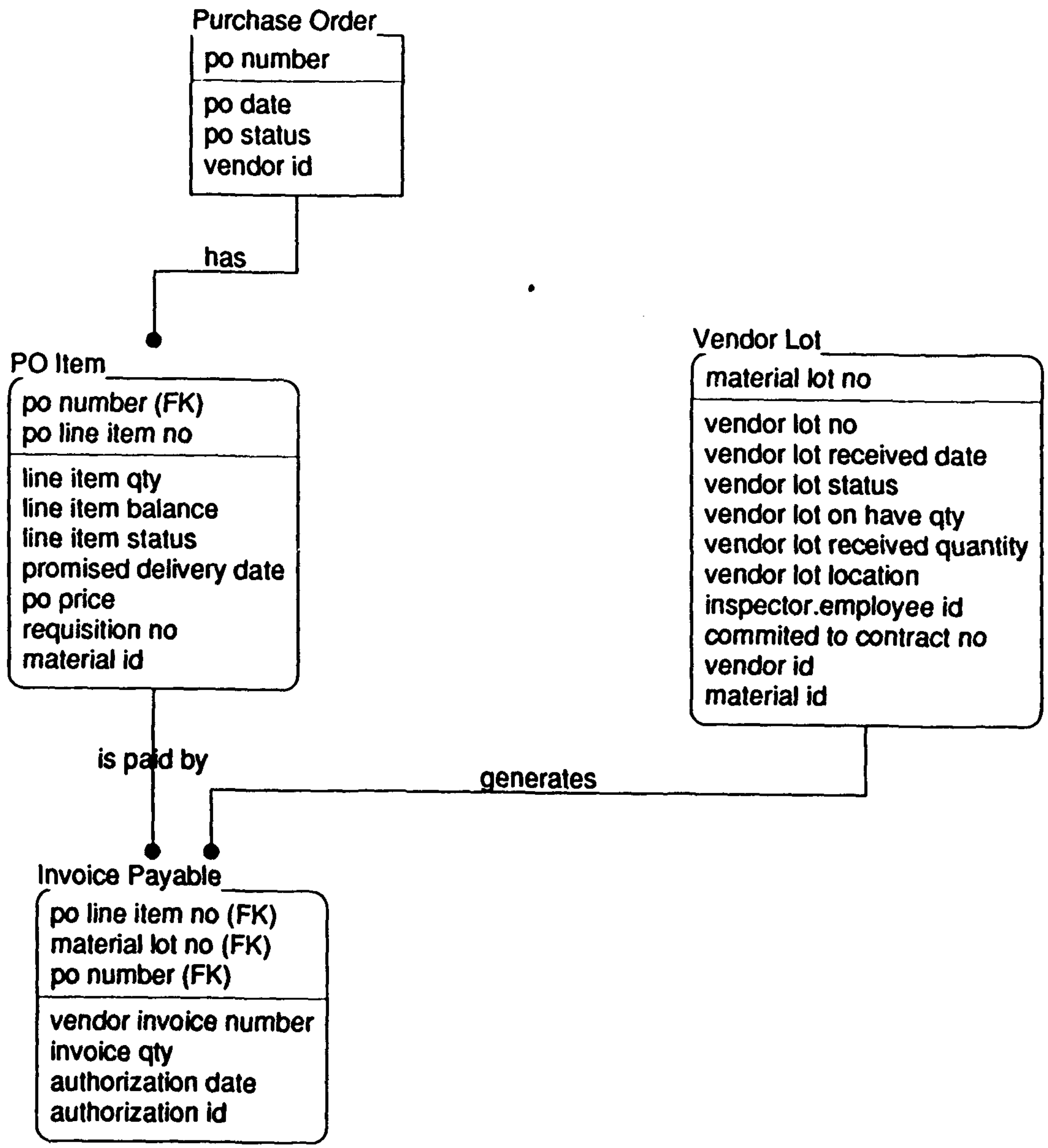
Manufacture Product



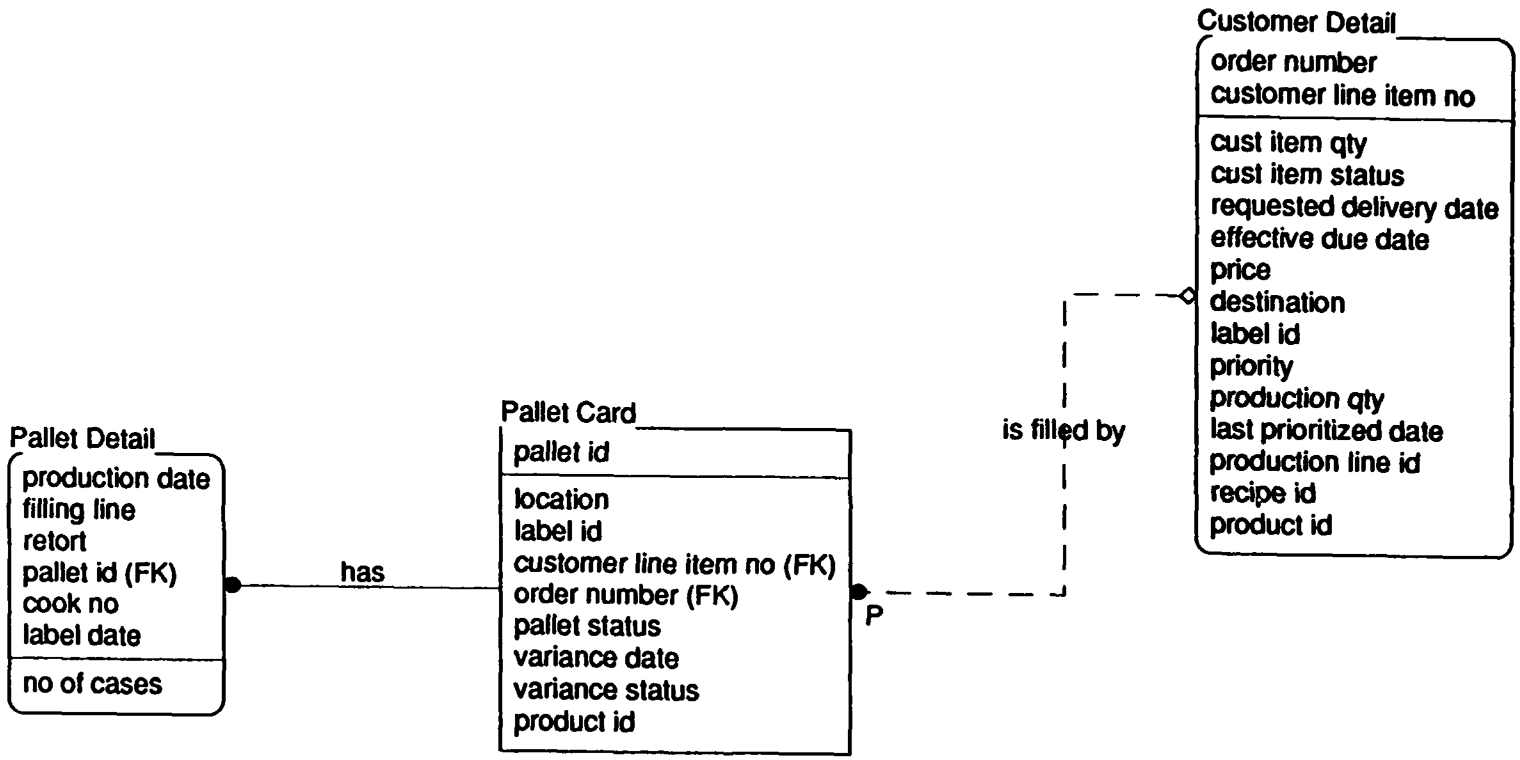
Iode A312 - Prepare And Issue Purchase Order



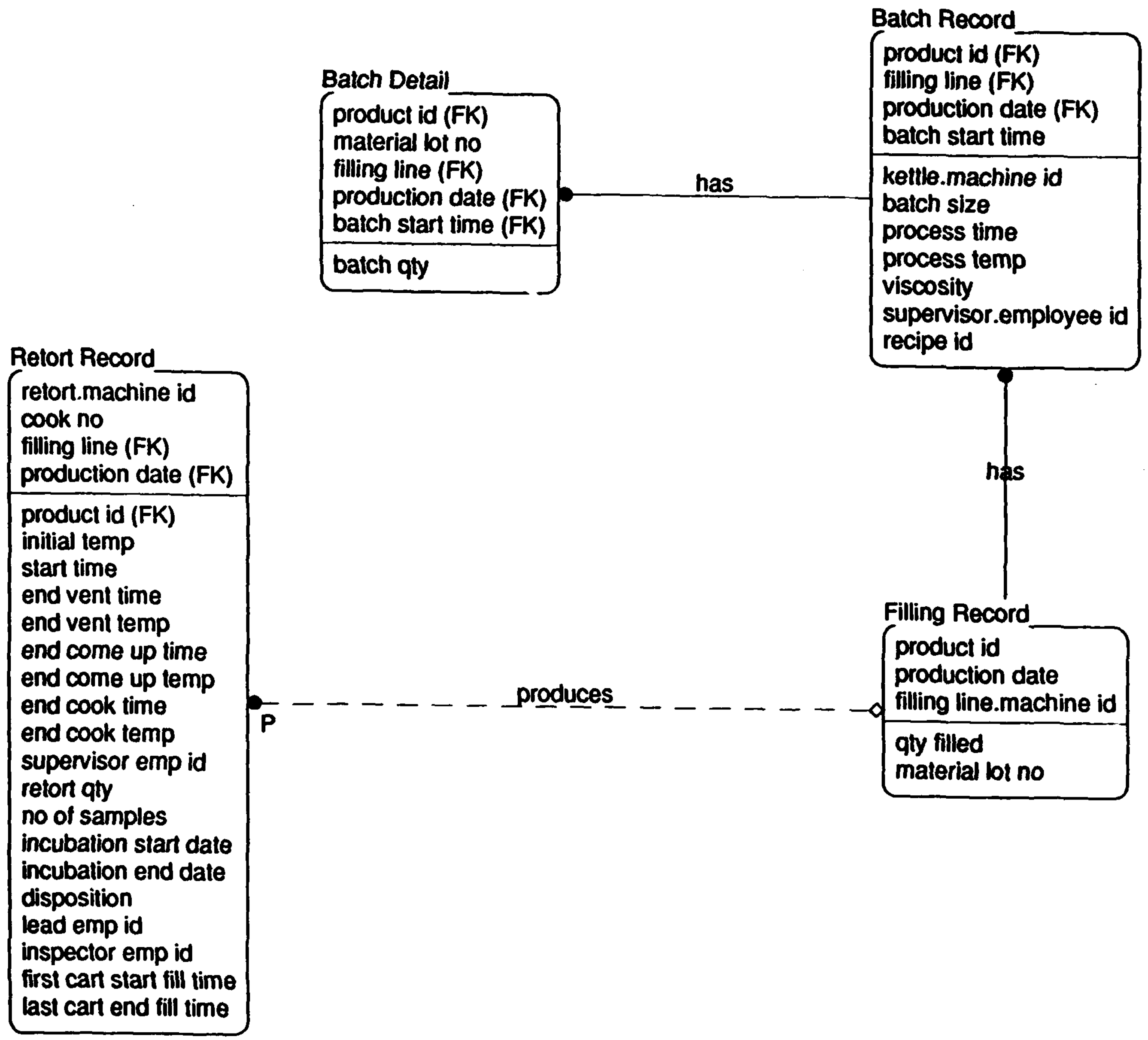
Node A313 - Receive, Inspect, and Store Shipment



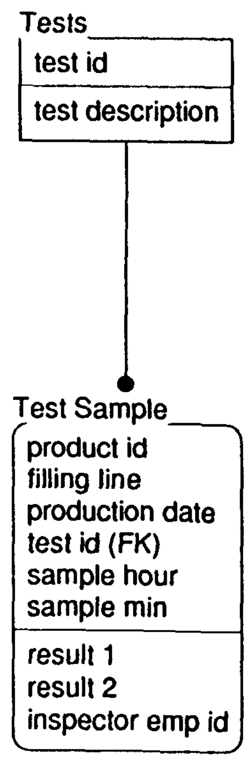
Node A314 - Update Accounting and Authorize Payment



Node A341 - Update Inventory Record



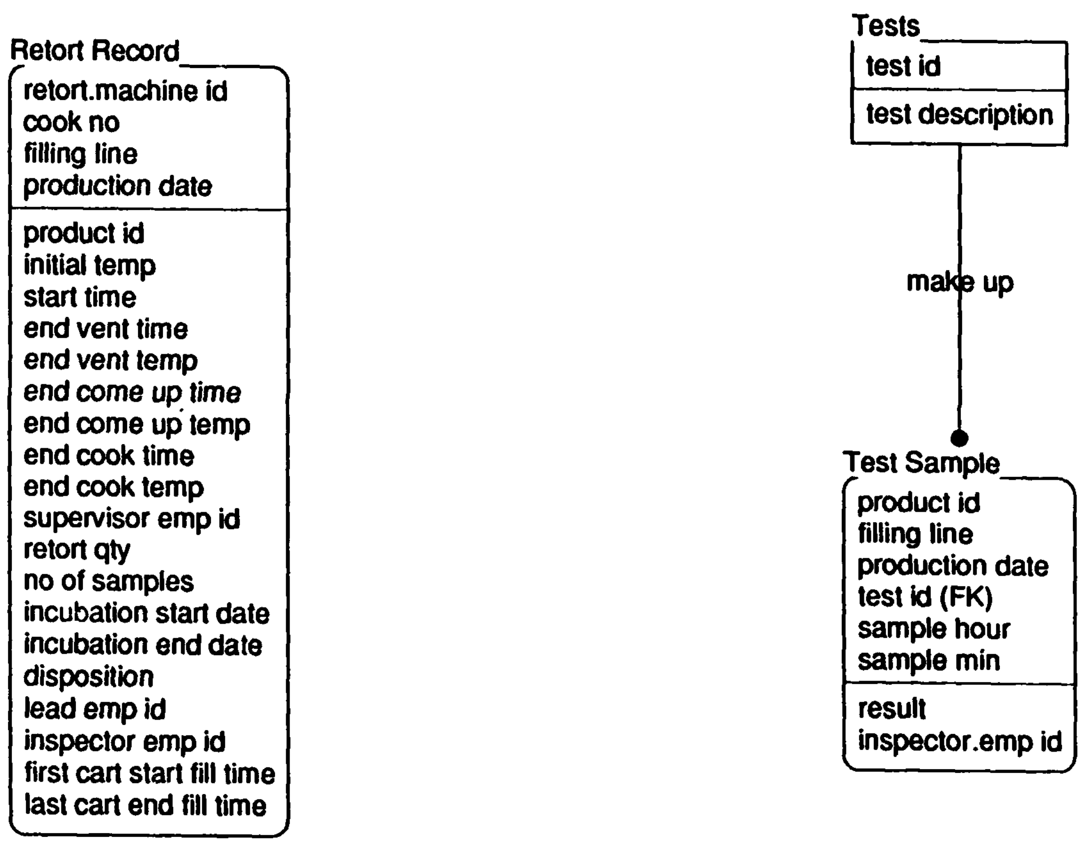
Node A342 - Update Processing Records



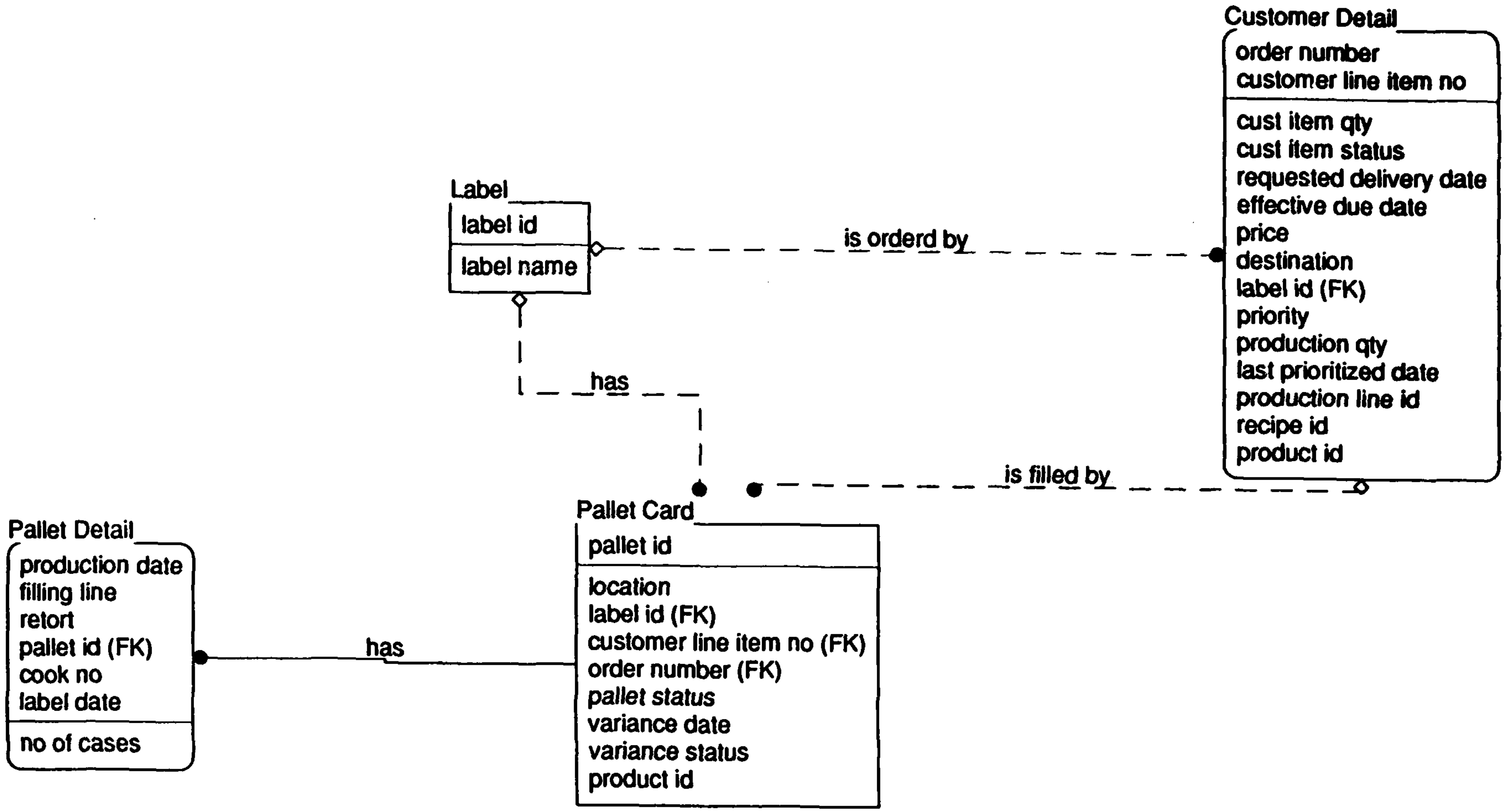
Node A343 - Update Inspection Records

Section IV

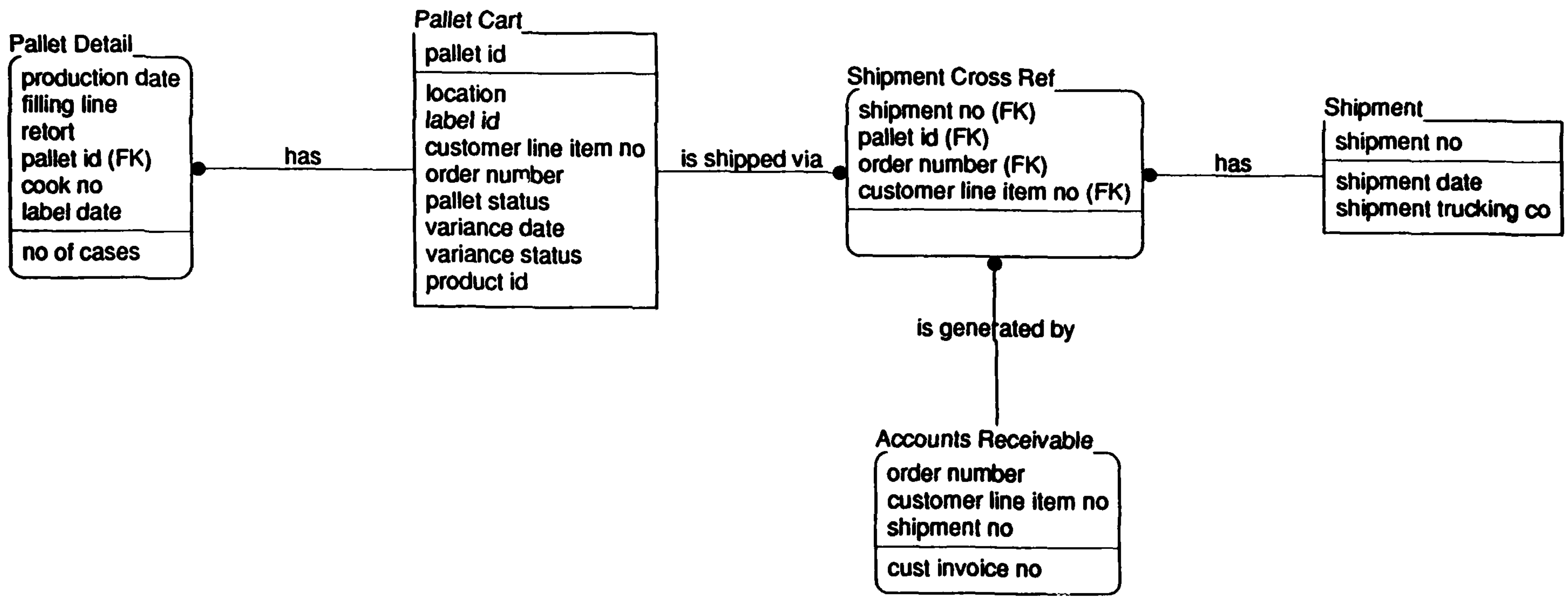
Control Manufactured Product



Node A41 - Perform finished Product Quality Assurance



Node A42 - Control Finished Goods Inventory



Node A43 - Ship Finished Product and Update Records

Section V

Summary

GLOSSARY

Accepted Quantity
The amount of material id that will be purchased from vendor id under the combination solicitation id, product id/recipe id in the Material Purchase Plan.

Actual Production Quantity
The quantity actually produced under the combination schedule id, machine id, order no., order line item no.

Actual Time
The actual time that machine id produced customer line item number of order number under schedule id.

Amount By Unit
The quantity of a specific material id used in a batch of a specific product id as stated by the product formula or recipe.

Amount End Units
The expected number of units of finished products per batch of the recipe or formula.

Authorization Date
The date on which purchasing authorizes accounting to pay the invoice.

Authorization ID
The id number of the purchasing agent who authorized payment of the invoice.

Avail Time
Total time available on machine id for period length.

Batch Quantity
The quantity of material lot number associated with the combination of product id, filling line, production date, batch start time.

Batch Size
The total gallons of ingredients batched.

Batch Start Time
The time of day at which a specific batching process was started.

Begin Effective Period
The starting date of the schedule identified by schedule id.

Break Price
The price per case of product over the range between break quantity limits.

Break Quantity
The sales quantity that defines the upper limit at which a break price applies.

Can Size
A container code, based on the quantity of product it can hold and the design specification of the container.

Can Spec. Material Id
The unique identifier of a particular container.

Committed to Contract Number
The contract or order number, if any, to which this material lot is specifically committed.

Cook Number
A unique identifier of a batch of product sterilized in the same retort at the same time on the same production date.

Cook Temperature
The actual batch processing temperature associated with the combination process id, filling line, production date, batch start time, kettle, machine id.

Cook Time
The actual batch process time associated with the combination process id, filling line, production date, batch start time, kettle, machine id.

Committed Time
The time committed on machine id by schedule id for production of customer line item number of order number.

Customer City
The city associated with customer id.

Customer FAX
The FAX number associated with customer id.

Customer ID
A unique identifier for a specific customer of the enterprise.

Customer Item Quantity
The number of cases of product id associated with the combination of order number, customer line item number.

Customer Item Status
An indicator of whether or not the customer line item number is open or closed.

Customer Line Item No.
A unique identifier of a line item on a customer order.

Customer Name
The name of a customer associated with customer id.

Customer Phone
The phone number associated with customer id.

Customer PO Number
The customer purchase order number associated with the enterprise order number.

Customer State
The state associated with customer id.

Customer Street
The street associated with customer id.

Customer Zip
The zip code associated with customer id.

Date In Service
The date on which the machine was placed in service.

Date Last Done
The last date of preventive maintenance task id on machine id.

Date Last Purchase
The date on the purchase order that was used for the latest purchase of a material.

Date of Work
The date on which employee id worked on customer line item number of order number.

Delivery Date
The desired delivery date on a request for quotation.

Department id
A unique number identifying a department or work unit of the enterprise.

Department Name
The name associated with Department id.

Department Phone
The phone number associated with Department id.

Destination
Shipping destination of customer line item number.

Disposition
An indicator of the final disposition of the cook.

Done By
The author of the most recent schedule revision.

Effective Due Date
The date production has to be completed in order to meet the requested delivery date of customer line item number of order number.

Employee City
The residence city associated with employee id.

Employee First Name
The surname associated with employee id.

Employee Hourly Rate
The rate of pay associated with employee id.

Employee ID
A unique identifier for each employee of the enterprise.

Employee Last Name
The family name associated with employee id.

Employee Phone
The residence phone number associated with employee id.

Employee State
The residence home state associated with employee id.

Employee Street
The residence street address associated with employee id.

Employee Zip
The residence zip code associated with employee id.

End Cook Temp
The temperature of the retort chamber at end cook time.

End Cook Time
The time at which the retort completes the sterilization process.

End Come Up Temp
The temperature of the retort chamber at end come up time.

End Come Up Time
The time at which the retort reaches its start cook temperature.

End Effective Period
The ending date of the schedule identified by schedule id.

End Vent Temp
The temperature of the retort chamber at end vent time.

End Vent Time
The time in the retort cycle at which the water has completely filled the retort chamber.

Filling Line
A unique identifier for a specific filling line in the production area.

Filling Line . Machine Id
A unique identifier for a specific filling line in the production area.

First Cart Start Fill Time
The point in time at which the first package off the filling line was placed in a retort basket for a cook no.

Frequency
The interval between successive preventive maintenance for task id.

High Limit
The high point of the acceptable range associated with a particular test id/material id combination.

Hourly Rate
The production rate, in cases per minute, associated with unique combinations of product id, recipe id, and production line id.

Hours
Employee hours expended by employee id on task id of work order id.

Hours Worked
The number of hours of date of work during which employee id worked on a customer line item number of order number.

Incubation End Date
Date on which incubation test ends.

Incubation Start Date
Date on which incubation test begins.

Initial Temp
The actual initial temperature of product associated with the combination retort.machine id, cook number, production date.

Inspector.Empl ID
The identifier for the individual in Quality Assurance responsible for quality testing.

Invoice Quantity
The amount of the material lot number for which the invoice applies.

Kettle Machine ID
A unique identifier for a specific kettle in the production area.

Label Date
The date on which the label was applied to the container.

Label ID
A unique identifier for a customer label specific to a product id.

Label Name
The description associated with label id.

Labor Quantity
The number of workers associated with the combination machine id, skill id.

Last Cart End Fill Time
The point in time at which the last package off the filling line was placed in a retort basket for a cook number.

Last Price Paid
The latest actual unit price paid for a material id.

Last Prioritized Date
Date on which the customer line item number of order number was last prioritized.

Last Revised Date
The date on which the schedule identified by schedule id was last revised.

Lead.Employee Id
Employee who pulls the samples of a cook number for incubation.

Lid Spec. Material Id
The unique identifier of a particular lid.

Line Item Balance
A quantity indicating the amount of the line item quantity that has not been satisfied by the vendor.

Line Item Quantity
The quantity associated with a unique combination of PO number/PO line item number.

Line Item Status
An indicator of whether a line item is closed out or still open.

Location
The location of pallet id in the finished goods storage area of the factory.

Low Limit
The low point of the acceptable range associated with a particular test id/material id combination.

Machine Description
A description name associated with machine id.

Machine ID
A unique identifier for a specific machine or system of equipment in the enterprise production facility.

Machine Location
The location of the machine in the production area.

Manufacturer
The company that built the machine.

Material Description
A name used to describe a material id.

Material ID
A unique number that identifies a specific material that is inventoried. The material is determined by its description and specification.

Material Lot Number
A unique number that specifies a material lot at the lowest level chosen by management. Ideally, this would be a surrogate for unique combinations of vendor id, material id, vendor lot no, vendor lot received date.

Material Units of Measure
The unit of measure in which material id is purchased and inventoried.

Move Quantity
The amount of the material lot number to be moved on the particular production date.

Move to Location
The location to which the material is to be moved.

Net Weight
Weight of the finished product container and its contents.

Number of Cases
The number of cases associated with the combination pallet id, cook number, retort, filling line, production date.

Number Required
The number of repair parts associated with work order id and material id.

Number of Samples
The number of packages taken from the retort for incubation testing.

Operation Sequence
A unique identifier of the sequence of the operation described by the combination production line id and process id.

Order Date
The date the order was taken.

Order Number
A unique identifier, assigned by the enterprise, for a customer order.

Original Date

The date on which the schedule identified by schedule id was first created.

Pallet ID

A unique identifier for a pallet of finished product.

Pallet Status

An indication of whether or not the pallet holds acceptable finished goods, finished goods on hold awaiting rework or variance, or rejected product awaiting disposition.

Period Length

The length of the schedule.

PO Date

The date of issue associated with PO number.

PO Line Item Number

The number which uniquely identifies a material, order quantity, and promised delivery date on the PO.

PO Number

A number that uniquely describes a specific purchase order.

PO Price

The price per unit associated with the combination PO number/PO line item number.

PO Status

An indicator whether or not the PO number is open or closed.

Price

The price per case of customer line item number of order number.

Price Valid Till

The date after which the current price schedule for the product is no longer valid.

Priority

A scheduling priority number determined by production planning.

Process Description

A description associated with process id.

Process Id

A unique identifier of a food [illegible]ess.

Process Temperature
The target batch temperature associated with a specific product id/recipe id combination.

Process Time
The target batch time associated with a specific product id/recipe id combination.

Product Due Date
The proposed date of start of production of a product id and solicitation id.

Product ID
A unique number that identifies a product, which includes the recipe and the container.

Product Name
A name associated with a unique product id.

Production Date
A date on which production takes place.

Production Line Description
A description associated with production line id.

Production Line ID
A unique identifier of a set of production processes.

Production Quantity
The amount of production required, considering normal reject rates, in order to meet the customer item quantity.

Promised Delivery Date
The date on which the vendor promised delivery of the line item.

Quantity Filled
Number of cases of product id filled on filling line machine id on production date.

Quantity Per Case
The packing quantity per case of the product.

Quote Date
The date the quotation was given.

Quote Expiration Date
The date of expiration associated with quote id.

Quote ID
A unique identifier associated with a quotation given to a customer.

Quote Quantity
The quantity associated with the combination of quote id, customer id, product id.

Quote Ship Date
The promised date of shipment associated with the combination of quote id, customer id, product id.

Quote Unit Price
The price per case of product associated with the combination quote id, customer id, product id.

Recipe ID
A unique identifier for a formulation for a given product id.

Recipe Size
The standard batch size for a particular combination of product id/recipe id.

Recovery Percent

The percent of material yield from raw material inventory to finished product.

Reorder Point
The quantity of on-hand plus on-order inventory by material id at which it is recommended that a replenishment be made.

Reorder Quantity
The recommended quantity of material id order when an order is placed.

Reply Price
The unit price associated with reply quantity.

Reply Quantity
The amount of material id quoted by vendor id as associated with the combination of solicitation id, product id, recipe id.

Request Quantity
The quantity of material id associated with the combination of solicitation id, product id, recipe id, vendor id.

Requested By Department Id
The department requesting the work order id.

Requested Delivery Date
The desired date of delivery of customer line item number of order number.

Requisition Date
The date on which a specific requisition number was prepared.

Requisition Number
A unique identifier for a request to have a material replenished.

Requisition Quantity
The amount of material associated with a specific requisition number.

Result
A conclusion drawn from the examination of a sample of production.

Retort
A unique identifier of a specific retort in the production area.

Retort,Machine ID
A unique identifier of a specific retort in the production area.

Retort Quantity
Number of units of product associated with a specific cook number.

Rework Rate
The percent of production that is non-conforming, but acceptable after rework.

Sales,Employee Id
The identifier of the sales person who took order number.

Sample Hour
The hour on production date at which product id was sampled on filling line machine id for test id.

Sample Minute
The minute on production date at which product id was sampled on filling line machine id for test id.

Schedule ID
A unique identifier for a production schedule.

Schedule Sequence
The position in the production schedule id of the particular customer line item number.

Selected Quantity
The quantity proposed for production of a specific product id by the enterprise in the response to solicitation.

Serial Number
The manufacturer's serial number for the machine.

Shift Number
The shift associated with date of work.

Shipment Date
The date on which shipment actually occurs.

Shipment Number
A unique identifier for a shipment of pallets.

Shipment Trucking Co
The carrier handling the shipment.

Skill Description
A description associated with skill id.

Skill ID
An identifier for skill level associated with employees of the enterprise.

Solicitation Completed By
The individual who completed the response to solicitation.

Solicitation Completed On
The date the response to solicitation is actually submitted.

Solicitation Due Date
The date by which the response to solicitation must be submitted.

Solicitation Due Time
The hour of the due date that the solicitation must be submitted.

Solicitation ID
A unique identifier for a contract solicitation, or request for proposals.

Solicitation Issue Date
The date that solicitation is made public.

Standard Cost
The cost per unit of material currently being used to establish the unit prices of products in which the material is used.

Standard Price
The current quotation price of the product per case.

Standard-Reject-Rate
A reject rate (percent loss) on finished product based on past experience.

Standard Shift Length Per Day
The number of hours of use for machine id in a standard shift.

Start Time
The time at which the retort cycle begins.

Status
An indicator of whether the requisition is open (not acted on) or closed (acted on).

Supervisor.Emp ID
A unique identifier of an employee with supervisory responsibility for an operation.

Target Cook Temperature
The target retort cook temperature associated with a specific product id/recipe id combination.

Target Cook Time
The target retort cook time associated with a specific product id/recipe id combination.

Target Initial Temperature
The minimum required initial temperature of product associated with a specific product id/recipe id combination, at the point in time that it goes into the retort.

Task Description
A description associated with task id.

Task Id
A unique identifier of a task for preventive maintenance or repair.

Test Description
A description of the procedure associated with the identifier test id.

Test ID
A unique number that identifies a specific test procedure for incoming materials.

Test Result
An accept or reject result associated with a unique combination of material lot number/test id/material id.

Transaction Type
An identifier field for each of the following: A movement from raw material into work in process or a movement from work-in process back to raw material.

Type Retort
Type of retort used.

Unit of Measure
The units of measure used for a material in the product formula or recipe.

Variance Date
The date a variable was requested for product on hold.

Variance Status
An indicator of whether or not the variance is granted.

Vendor City
City associated with vendor id.

Vendor Contact Name
The individual salesman normally contacted at the company indicated by vendor id.

Vendor FAX
The FAX number associated with vendor id.

Vendor ID
A unique identifier for each qualified vendor.

Vendor Invoice Number
A number that identifies the vendor's invoice for materials delivered.

Vendor Lot Location
The location in which the material lot no is currently stored.

Vendor Lot No
A number that identifies the vendor production lot from which the material originated; typically the Julian date of production.

Vendor Lot on Hand Quantity
The amount of the material lot number in inventory.

Vendor Lot Received Date
The date on which the material lot no was received at the enterprise.

Vendor Lot Received Quantity
The amount of the material lot originally received into inventory.

Vendor Lot Status
A classification of each material lot into accept, reject, hold based on quality control requirements.

Vendor Name
The company name of an approved supplier of material.

Vendor Phone
The phone number associated with vendor id.

Vendor State
State associated with vendor id.

Vendor Street
The street address associated with vendor id.

Vendor Type
An indicator whether or not the vendor id is a small business.

Vendor Zip
The zip code associated with vendor id.

Viscosity
The viscosity of batched ingredients.

Work Order Due Date
The requested completion date of work order id.

Work Order Id
A unique identifier of a work order to perform either preventive maintenance or repair.

Work Order Type
The type of work order; i.e., preventive maintenance or repair.

COMBAT RATION
ADVANCED MANUFACTURING TECHNOLOGY DEMONSTRATION
(CRAMTD)

Simulation Model
Software Requirements Specification
Version 1.0
Technical Working Paper (TWP) 15

Mohsen A. Jafari
Veenkivarum S. Srinivasan
Industrial Engineering Department
Rutgers University
October, 1990

Sponsored by:
DEFENSE LOGISTICS AGENCY
Cameron Station
Alexandria, VA 22304-6145

Contractor:
Rutgers, The State University of New Jersey
THE CENTER FOR ADVANCED FOOD TECHNOLOGY*
Cook College
N.J. Agricultural Experiment Station
New Brunswick, New Jersey 08903

DR. JACK L. ROSSEN
Program Director, CRAMTD
DR. JOHN F. COBURN
Associate Director, CRAMTD

TEL: 908-932-8306
FAX: 908-932-8690

TABLE OF CONTENTS

1.0 General

1.1 Purpose of the Software Requirements Specifications

STP #4 of Contract DLA 900-88-D-0383 between Rutgers University and the Defense Logistics Agency requires that the contractor design and code a simulation model which will be capable of demonstrating steady-state performance of the projected CRAMTD production system at level 3 automation. The purpose of this document is to provide the software requirements specification to guide the design and coding process. This document is a working document and is subject to revision as the project proceeds.

1.2 Project References

There are two documents which describe the cooking, filling and sealing requirements of an automated tray pack production line.

1. Sigethy, A., Descovich, T., and Boucher, T. O. "Revised Automation Control Strategy for Tray Pack Filling/Sealing Line", CRAMTD Technical Working Paper (TWP)14.

2. Boucher, T. O., Sigethy, A., Bruins, R., and Gursoy, M. "Revised Level 1 Automation Control Strategy for CRAMTD Cooking Operation", CRAMTD Technical Working Paper (TWP)15

There is one document which describes the commercial simulation language being used for this project:

1. Pegden, D. C., Introduction to SIMAN, Systems Modeling Corp., 504 Beaver Street, Sewickley, PA 15143.

Other related documentation are as follows:

a) Technical Proposal: STP #4 Under Contract DLA 900-88-D-0383.

b) Previously developed technical documentation relating to this project: None

c) Significant corresponds related to project: None

d) Documentation concerning related projects: None

e) Manuals or documents that constrain or explain technical factors affecting project development: None

f) Standards or reference documentation:

1) Documentation Standards and Specifications: None

2) Programming conventions: Introduction to SIMAN

3) DoD or Federal Standards: None

4) Hardware Manual: None

1.3 Terms and Abbreviations

CRAMTD - Combat Rations Advanced Manufacturing Technology Demonstration.

2.0 System Summary

2.1 Background

The major task to be performed under the CRAMTD project is to bring together existing advanced food manufacturing technology and to develop new technology that will enable the design of food production systems that are flexible, cost effective, and capable of producing products of high quality. In order to analyze such a system in the design phase as well as in the operational phase, it is important to develop a simulation model of the system. Such a simulation model is capable of analyzing the performance of the system under different production planning/control policies, system layouts, scheduling rules, and so on. As a by-product of this development effort, it is also

possible to determine the benefits of using advanced food manufacturing technology as opposed to labor intensive manufacturing methods.

Two types of food production lines are under consideration for simulation: Tray-pack line (See Figure 1) and MRE pouch line (see Fig. 2). For each line, a simulation model is developed. These models are written in SIMAN (a commerically available simulation language) and are limited to the flow of different materials through different processes that exist in the production lines.

2.2 Objectives

There are three objectives to be satisfied by the simulation models. These objectives are as follows:

1. To compute different performance measures of the system, such as:

a. Production rate of tray pack or MRE pouch production line.

b. Average system flow time for a tray pack or a MRE pouch.

c. Utilization of different stations in the tray pack or MRE pouch production lines.

d. Queueing characteristics for each station in the tray pack or MRE pouch production lines.

2. To study different scheduling policies for daily production.

3. To provide a graphical representation of the production line operation.

2.3 System Definition

Since simulation programs are written in SIMAN, it is important that SIMAN Software organization is defined together with the user interface. Figure 3 (ref. Introduction to SIMAN) represents SIMAN Software organization and Figure 4 represents the user perspective. A SIMAN simulation is divided into three distinct activities: System model development, experimental frame development, and data analysis. Within these three activities, the SIMAN Software consists of five individual processors (model, experimental, link, run, output) which interact through four data files:

2.3.1 Model File Generation: The model processor is used to construct a block diagram representing specific process functions. The data file that is generated is called the model file. This file may be generated in an interactive graphics mode through an editor called "BLOCKS".

2.3.2 Model File Compilation: Model file is compiled into an appropriate format to be read by link processor. This is done through "MODEL" compiler.

2.3.3 The Experimental File Generation: The experimental processor is used to define the experimental frame (containing all the input parameters) for the system model. The data file that is generated is called the experiment file. This file may be generated in an interactive graphics mode through an editor called "ELEMENTS".

2.3.4 Data File Compilation: Input data file is compiled into an appropriate format to be read by Link processor. This is

done through "EXMPT" Compiler.

2.3.5 Program Generation: *The Link processor combines* the model file and the experiment file to produce the program file.

2.3.6 Running Simulation: The program file is input to the run processor which executes the simulation runs and writes the results into an output file.

2.3.7 Processing the simulation Output: the output processor is used to analyze format and display the data contained in the output file.

2.3.8 CINEMA MODEL: This is a graphical simulation. Different entities (work-stations, material, etc.) are represented by different icons and stored in a file through a graphics interface. The dynamics of these icons are governed by SIMAN program. The result will be animation of the system operation on the screen.

2.3.9 The block diagram or model file consists of six overlapping modules:

2.3.9.1 Scheduling Module - This contains the scheduling routine for daily production.

2.3.9.2 Resource Selection Module - The resources (work stations) required to produce a given product are selected in this module, and consequently turned on.

2.3.9.3 Material Transportation Module - This contains the routine for material transportation between resources. It also checks the availability of material at each resource.

2.3.9.4 Filling Line Module - This module contains instructions relating to the simulation of filling and

sealing operations.

2.3.9.5 Packaging/Inspection Module - The instructions relating to the simulation of packaging and inspection stations are included in this module.

2.3.9.6 System Reset Module - All the system parameters are reset and initialized.

SIMAN SOFTWARE ORGANIZATION

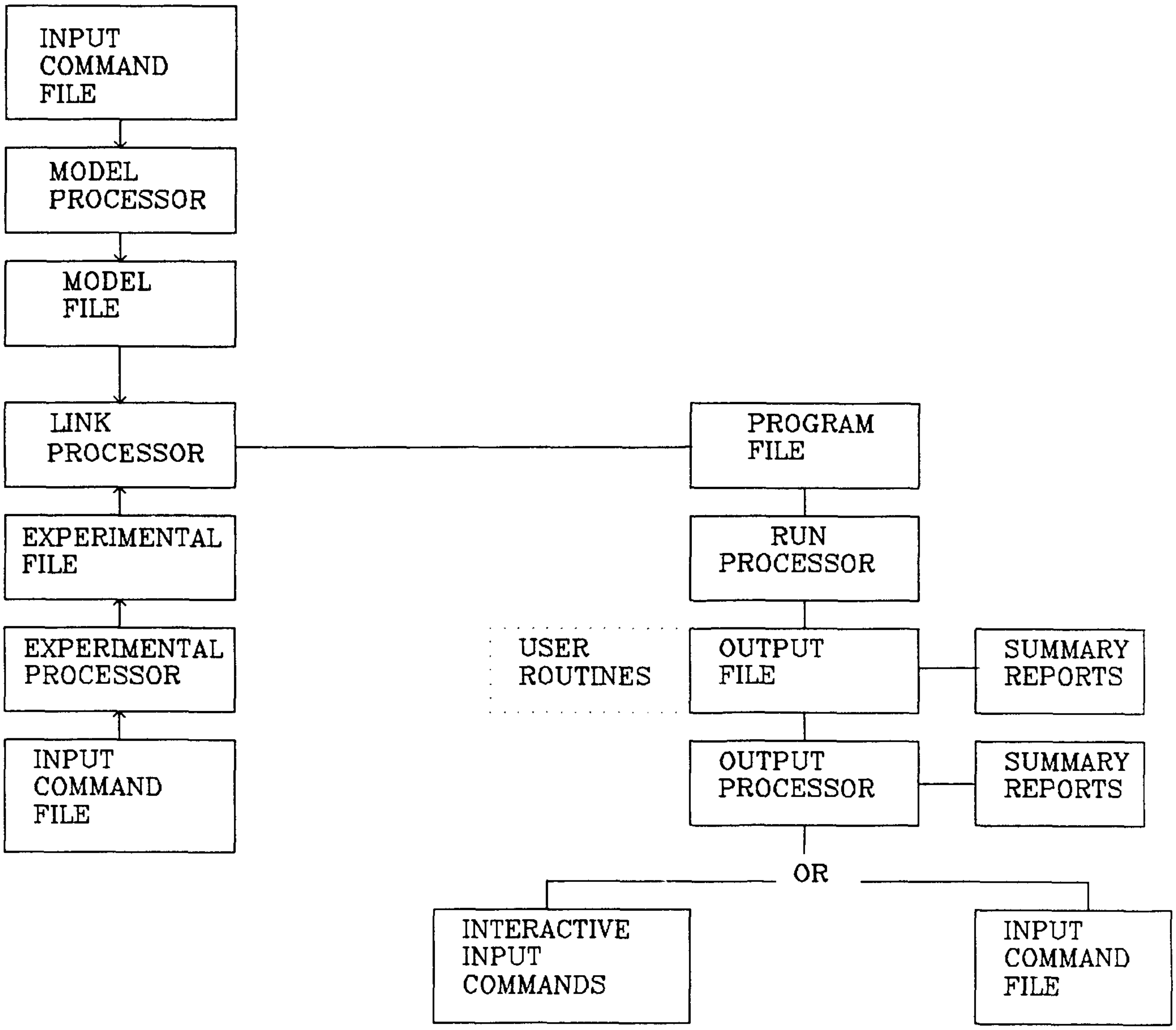


Fig 3 SIMAN SOFTWARE ORGANIZATION

2.4 SYSTEM DIAGRAM (USER PERSPECTIVE)

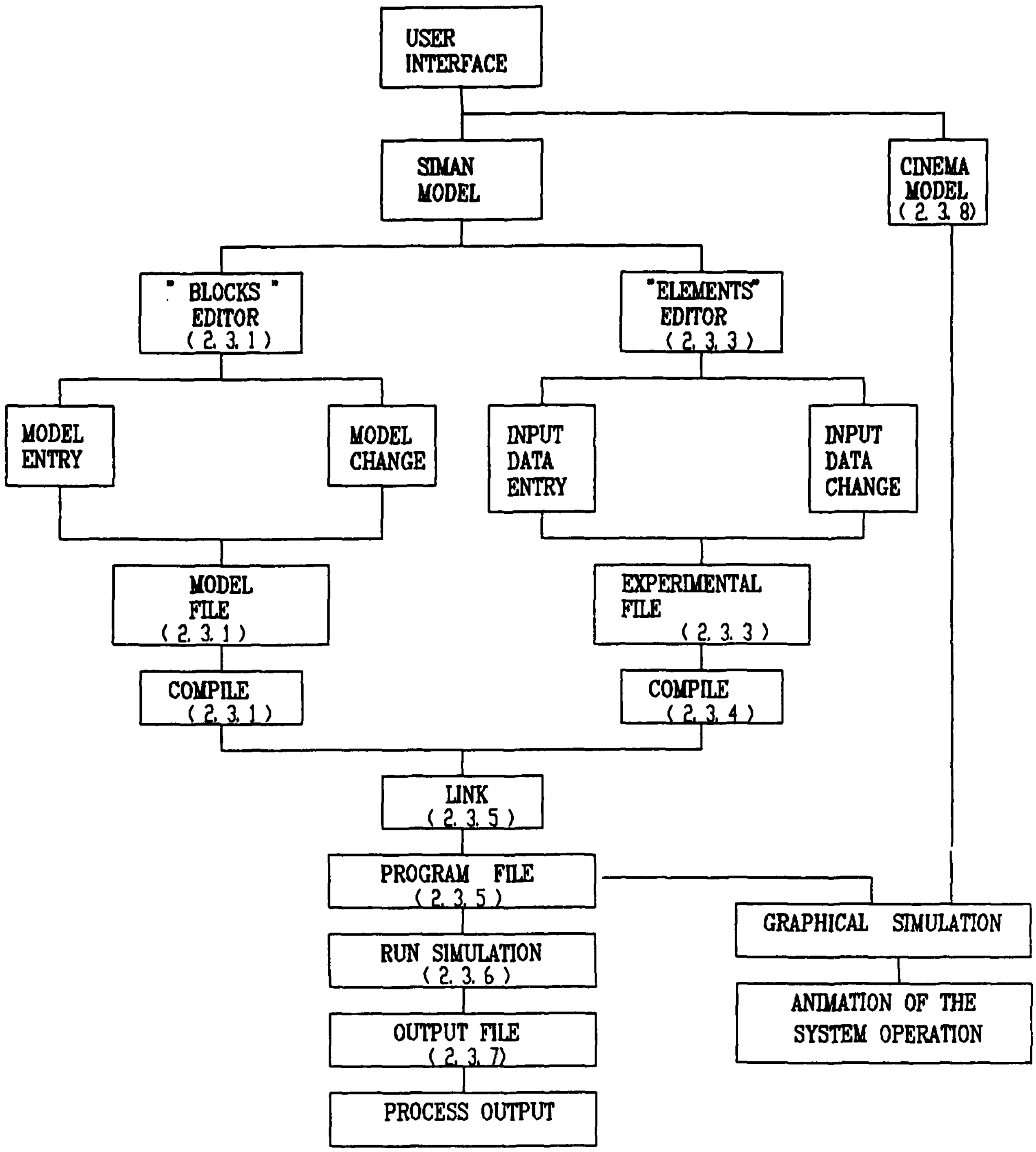


FIG 4: SOFTWARE ARCHITECTURE - USER PERSPECTIVE

2.5 Computer Program Identification

2.5.1 The following programs belong to SIMAN:

1 - "BLOCKS" Editor
2 - "ELEMENTS" Editor
3 - "MODEL" compiler
4 - "EXMPT" compiler
5 - LINK
6 - RUN

2.5.2 A user application program consists of two files:

2.5.2.1 A file containing block diagram.

2.5.2.2 A file containing input parameters.

2.6 Assumptions and Constraints

None identified.

3.0 Environment

3.1 Equipment Environment

a) IBM PS2, DOS Operating System, 640K RAM

b) Storage Media 3.5" Floppy disk

c) Output devices VGA monitor, Printer

d) Input devices keyboard and mouse

e) No additional communications requirements

3.2 Support Software Environment

Simulation program is developed in SIMAN Simulation language which requires DOS operating system and MICROSOFT FORTRAN compiler, version 4.0.

3.3 Interfaces

None.

3.4 Security and Privacy

No requirement for security and privacy.

4.0 Detailed Characteristics and Requirements

4.1 Specific Performance Requirements

4.1.1 Accuracy and Validity

a) The accuracy of final results depends on the length of simulation run. As long as the simulation clock does not exceed $1x10^9$ time units (time unit is implicitly defined through the input data), the accuracy is up to 5 digits after the decimal point.

b) Since simulation by itself is an approximation, it is necessary to validate the results. There are several validation techniques existing in the simulation literature (ref. Discrete-Event System Simulation by Banks and Carson).

c) User is responsible for insuring the accuracy of input data as required by SIMAN. These requirements are specified in Introduction to SIMAN.

d) No data transmission checks are required.

4.1.2 Timing

a) Throughput time of a simulation program depends on the complexity of the system being simulated as well as the length of the simulation run. The latter may be specified in terms of either the simulation clock or some counter (counting the number of some events taking place in the system).

b) Response time to queries and to updates of data files: in order of seconds.

c) Response time of major functions: for SIMAN function, the response time is in order of seconds.

d) Sequential Relationship of functions: SIMAN functions must be performed in an order as described in ref. Introduction to SIMAN.

4.2 Computer Program Functions

4.2.1 Identification of Computer Program No. 1

Program Name: BLOCKS ["*". MOD].

This program is used for entering, modifying, saving and retrieving model file ["*". MOD] which contains the block diagrams (2.3.1). "*" is a user specified file name.

4.2.2 Identification of Computer Program No. 2

Program Name: ELEMENTS ["*". EXP].

This program is used for entering, modifying, saving and retrieving experiment file. ["*". EXP] which contains input parameters (2.3.3). "*" is a user specified file name.

4.2.3 Identification of Computer Program No. 3

Program Name: MODEL ["*". MOD].

This program is used to compile the model file and create ["*". M] file (2.3.2).

4.2.4 Identification of Computer Program No. 4

Program Name: EXPMT ["*". EXP].

This program is used to compile the experiment files and create ["*". E] file (2.3.4).

4.2.5 Identification of Computer Program No. 5

Program Name: LINK ["*". M] ["*". E].

This program is used to link the compiled model and experiment file to generate program file ["*". P] (2.3.5).

4.2.6 Identification of Computer Program No. 6

Program Name: SIMAN ["*". P].

This program executes the program contained in program file (2.3.6) and generates an output file containing

simulation results.

4.2.7 Identification of Computer Program No. 7

Program Name: OUTPT [output file name].

This program is used to manipulate the simulation output file.

4.3 Inputs-Outputs

Already covered in Section 4.2 of this specification.

4.4 Data Characteristics

Input data must be specified according to the rules given in ref. Introduction to SIMAN. Resulting computations shall be in integers or single precision numbers displayed in decimal format or exponential format.

SIMAN required about 7.9 mega bytes of disk space and about 590K bytes of RAM in order to run. The user defined programs (files) require about 1 mega byte of disk space.

4.5 Failure Contingencies

This software is not critical to other system operations and will not have redundancy or fail safe provisions. Failure during operation will result in the loss of files not saved. Failure will require a restart and lost files will have to be reloaded.

4.6 Design Requirements

The only design requirements are those specified by SIMAN, (ref. Introduction to SIMAN).

4.7 Computer Security Requirements

None.

4.8 Human Performance Requirements

Knowledge of simulation methodologies as well as SIMAN is required.

5.0 Test and Quantification Requirements

There are several techniques in simulation methodology to test the validity of simulation results:

- Face Validity: Simulation results are examined by a person(s) knowledgeable about the system (being modeled).

- Sensitivity Analysis: Simulation outputs are observed by changing the value of input parameters.

- Input/Output Transformation: Under the same values for input parameters, outputs from the simulation model are matched against those obtained from the system (being modeled). This test is possible if the modeled system exists. To the extent possible, the simulation results will be compared against those obtained from the actual system installed in the CRAMTD demonstration site.

6.0 Notes

The following documents are cited in this specification, and can be made available to the reader to assist in understanding this specification.

Reference:

1. Boucher, T. O., Sigethy, A., Bruins, R., and Gursoy, M., "Revised Level 1 Automation Control Strategy for CRAMTD Cooking Operation", CRAMTD Technical Working Paper (TWP)15.

2. Sigethy, A., Descovich, T., and Boucher, T. O., "Revised Automation Control Strategy for Tray Pack Filling/Sealing Line", CRAMTD Technical Working Paper (TWP)14.

3. Pegden, D. C., Introduction to SIMAN, Systems Modeling Corp., 504 Beaver Street, Sewickley, PA, 15143.

4. Banks, J. and Carson, J. S., Discrete-Event System Simulation, Prentice Hall, 1984.

STP#1 -- TRAY PACK CONVEYOR

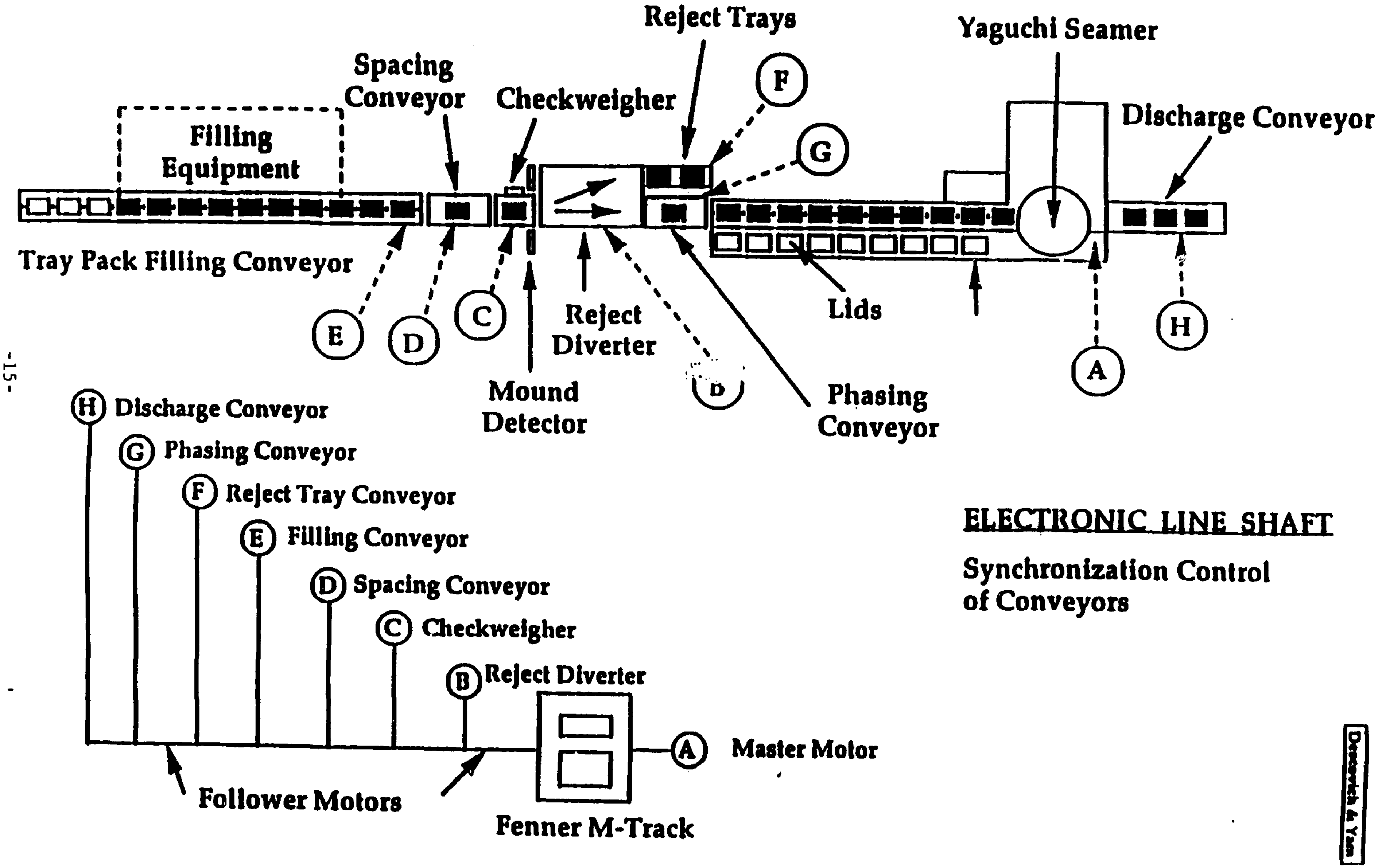
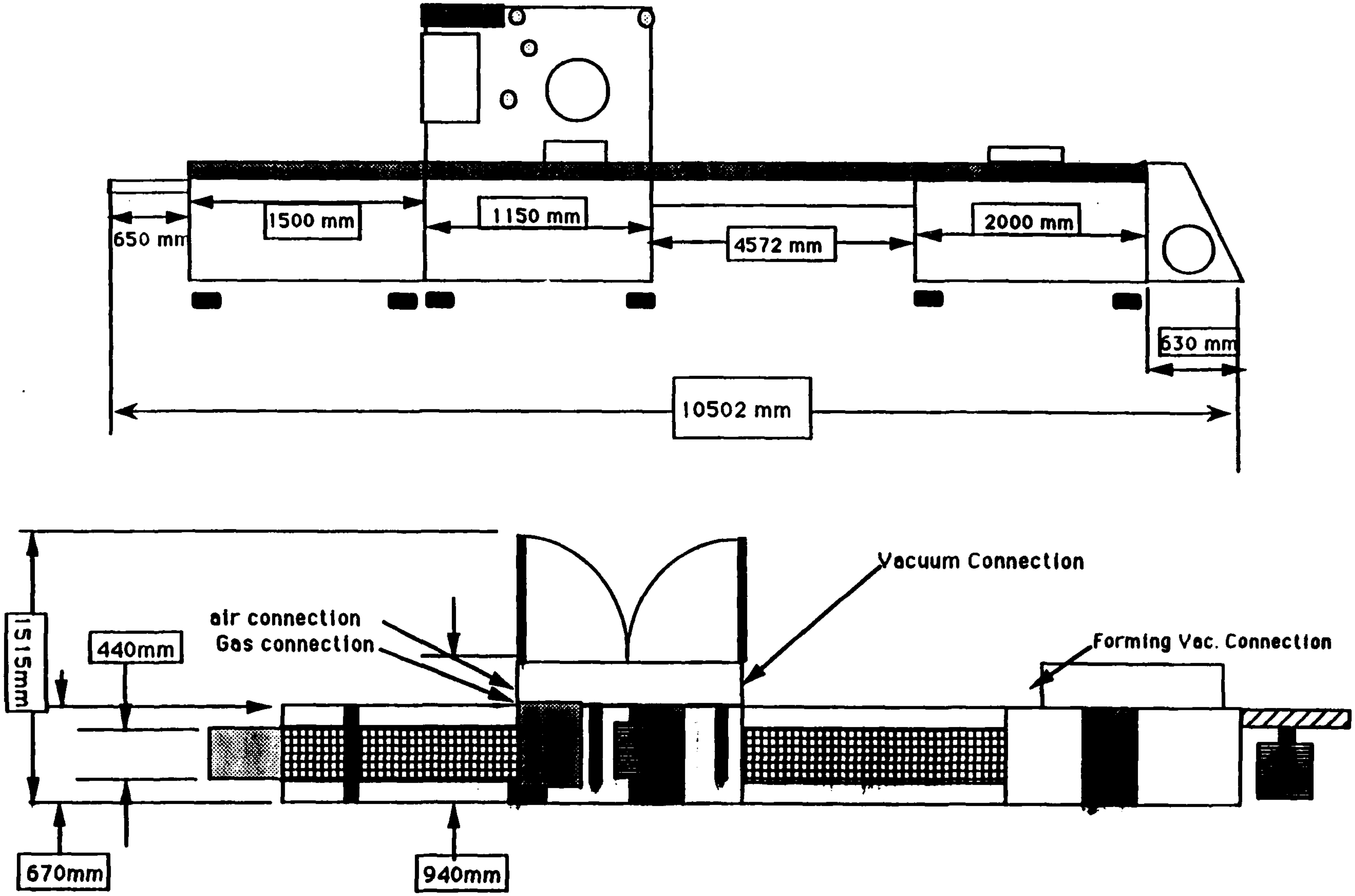


Fig. 1 - Tray Pack Conveyor

9/19/90
"RUTGERS"

3000/440



NO SCALE ,Dimension approxamate

F[illegible] 2: [illegible]/Fi[illegible]eal [illegible]ch [illegible]ine

COMBAT RATION ADVANCED MANUFACTURING TECHNOLOGY DEMONSTRATION (CRAMTD)

Report on CRAMTD Tray Pack and MRE Pouch Simulation Models
Technical Working Paper (TWP) 33

M.A. Jafari and V.S. Srinivasan
Industrial Engineering Department
Rutgers University
October 1991

Sponsored by:
DEFENSE LOGISTICS AGENCY
Cameron Station
Alexandria, VA 22304-6145

Contractor:
Rutgers, The State University of New Jersey
THE CENTER FOR ADVANCED FOOD TECHNOLOGY*
Cook College
N.J. Agricultural Experiment Station
New Brunswick, New Jersey 08903

DR. JACK L. ROSSEN
Program Director, CRAMTD
DR. JOHN F. COBURN
Associate Director, CRAMTD

TEL: 908-932-7985
FAX: 908-932-8690

*A New Jersey Commission on Science and Technology Center

1. Introduction

This technical report responds to task ref. number 3.7.3 of STP #4 under contract No. DLA 900-88-D-0387. It describes the discrete event computer simulation models developed for CRAMTD Advanced Tray-pack and MRE-pouch lines as part of STP#4: Design and Development of CIM Architecture for Food Manufacturing. These simulation models are capable of demonstrating steady-state performance of the projected CRAMTD production system at level 2 automation. They particularly include equipment utilization, production rate, queueing and inventory characteristics of the CRAMTD lines. The simulation models are implemented on a commercially available simulation language, namely SIMAN.

The outline of this report is as follows: In Section 2, we give an overview of simulation modeling, and in Sections 3 and 4, we describe the simulation models for the CRAMTD Advanced Tray-pack and MRE Pouch lines. In Section 5, we give guidelines for using the simulation programs. The simulation codes are given in Appendices I and II.

2. An Overview of Discrete Event Simulation Modeling

Discrete event computer simulation (will be referred to as simulation, hereafter) has been used by many researchers and practitioners to study complex systems. Though simulation can be used for both deterministic and stochastic systems, the common practice has been to use simulation for stochastic systems. In a typical stochastic system there are one or more sources of randomness. For instance, in a production line processing times on different machines can be randomly distributed according to some distribution function. Another example could be random error or noise inherent in almost any mechanical equipment. For instance, a typical filling machine has some degree of inaccuracy which is of a random nature.

One of the main reasons for building a simulation model is to be able to make some

inferences about the system based on the experiments performed on the model. This, however, requires that the model accurately represents the modeled system within some level of abstraction. This means that the simulation model needs to be validated prior to its utilization as a representation of the real system. Facilitation of simulation validation is one among several reasons to use special purpose simulation languages for computer programming. We will discuss simulation validation in more detail later. Next, we briefly describe the steps to simulation modeling. In a subsequent section, we will describe SIMAN.

Simulation Methodology

Computer simulation modeling is performed in several steps (ref. *Discrete-Event System Simulation*). The first step involves problem statement. In this step the system as a whole and its components must be clearly defined and the system boundary be recognized. In the case of CRAMTD lines we performed this step through meetings with the CRAMTD staff knowledgeable about the advanced Tray-pack and MRE Pouch lines. We also made our own observations about these systems.

The second step in simulation modeling is data collection. Two types of data may be collected: One type of data relates to what will be used as input data for the simulation model. The second type of data would serve as reference data for validation. We note that for a new system in the design stage, the data collection may not be possible mainly because the system does not exist and therefore does not have any history. In such a case, it may be possible to use data from a similar existing system. For the CRAMTD lines, we collected the first type of data through different channels: the CRAMTD staff and equipment vendors. We also used some of the historical data provided to us by Tray-pack and MRE pouch manufacturers. The latter was data related to product demand, shipping schedule, and casing line operation.

The third step in simulation modeling is to develop a conceptual model of the system. This model could be an informal model written in English-Like statements, or it could be more structured like a flow chart or block diagrams. In either case, this model would describe the logic behind the system operation and interaction that exist among different components of the system. It is desirable to verify this model. That is to say, compare it with the actual system prior to proceeding to the next step. For the CRAMTD lines, we developed block diagrams.

Given that the conceptual model accurately represents the system under study, the next step is to develop the operational model or the computer code from it. Two types of computer programming languages have been used for simulation: high-level programming languages, such as C, FORTRAN, PASCAL, and special purpose simulation languages, such as GPSS, and SIMAN. Each class of languages has its own advantages and disadvantages. General purpose languages are quite flexible and are easily portable from one hardware platform to another. However, simulation modeling using these type of languages usually requires quite a long development cycle. Because of the built in functions and procedures, the special purpose simulation languages lend themselves to a shorter program development cycle. However, these languages are less flexible and less portable.

There are two general approaches to simulation modeling: process- interaction approach and event-scheduling approach. In the former one, the modeled system is viewed in terms of its entities, the interaction between the entities, and the undergoing processes for each entity. In the event-scheduling approach the system is viewed in terms of its primary events which, when they occur, change the state of the system. The former approach is often employed by the special purpose simulation languages, whereas, the latter approach is more suitable for general purpose high level languages. The process-interaction approach happens to be a more natural modeling approach. It also facilitates

the validation procedure. These are the two major advantages of special purpose simulation languages over the general purpose programming languages.

Despite these advantages, the special purpose simulation languages, particularly, in micro-computer environment have a stringent memory constraint. The problem comes from the modeling approach associated with these languages. Each physical entity of interest in the real system (e.g., trays or pouches) is modeled by a logical entity requiring a computer memory space. Thus, the memory requirement for a simulation program grows very rapidly as the number of concurrent entities in the system grows. Nevertheless, this problem could be contained to a large extent by careful programming.

For the CRAMTD simulation models, we have used a special purpose simulation language, SIMAN, which is also commercially available. SIMAN has gained remarkable popularity, particularly among those who model manufacturing systems. Later in this report we will briefly discuss SIMAN.

Given the simulation code, the next step is to validate the model. There are several different approaches to model validation, some being somewhat subjective and others being more objective. One subjective approach is so called "face validation". Here, the simulation model and its results are examined by somebody who is quite familiar with the real system. One other subjective validation is through visual inspection of the simulation model operation. With the new special purpose simulation languages it is possible to develop a graphical model to animate the system operation. The inspection of this graphical model is another way of validating the simulation model.

A more objective approach for validation is called "input-output transformation". Here, the output of the simulation model is statistically compared to the data collected from the real system. It is, of course, important that the input conditions for the model match those of the real system at the time of data collection. The use of input-output

transformation is possible only in cases where the modeled system exists and output data could be collected. Another objective validation meth is "sensitivity analysis" where the simulation outputs are examined by changing the inputs. In sensitivity analysis we are not very much concerned with the absolute values of the simulation output. What matters is the direction by which a simulation output changes as we change one or more simulation inputs. This type of validation is very common and could easily be implemented.

For the CRAMTD simulations we were not able to perform input-output transformation mainly because the actual physical systems do not yet exist in their entirety. The types of validation that we have done are face and visual validation and sensitivity analysis.

In regard to sensitivity analysis, we have tested the sensitivity of the CRAMTD Advanced Tray-pack line production rate as the line speed and failure rate for the seamer change. As for the MRE Pouch line, the sensitivity of the production rate was tested against the line speed.

Having the validation procedure completed successfully, the simulation model may now be utilized for experimentation. One purpose for which the results from the CRAMTD simulation are used is NCIC as part of the CRAMTD base project.

A Brief Description of SIMAN

A SIMAN simulation is divided into three distinct activities: System model development, experimental frame development, and data analysis. Within these three activities, the SIMAN software consists of five individual processors (model, experimental, link, run, output) which interact through four files: model file, experimental file, program file, and output file. The model file is developed by a user and contains the simulation model (also called block diagrams). This model is compiled by the model processor into a format to be read by the link processor. The experimental file contains input data values for the

model. This file is compiled by the experimental processor into a format to be read by the link processor. The model and experimental files are linked together to make the program file. This file is executed using the run processor generating output data which are stored in output files. If desired, this file may be processed by the output processor.

SIMAN also has a graphical simulation capability. Here, different entities (workstations, raw material, packaging material, etc.) in the system are represented by different icons. These icons are stored in a file through a graphics interface. The dynamics of these icons are governed by the SIMAN simulation program. The result will be animation of the system operation on the screen. For more on SIMAN, see ref. *Introduction to SIMAN*, and *Simulation Model Software Requirements Specification, Version 1.0.*

3. Simulation of the CRAMTD Advanced Tray-Pack Line

First, we give a brief introduction to the CRAMTD Advanced Tray-pack line. Then, we describe the simulation model for this system.

The CRAMTD Advanced Tray-Pack Line

The CRAMTD Advanced Tray-Pack line is a hypothetical production line based on the technology being used in the CRAMTD pilot plant and proposed for use in the CRAMTD phase II. It is composed of four major areas: the cooking area, the filling area, the retort area, and the casing area. A schematic of the CRAMTD Advanced Tray-pack line is shown in Figure 1. The cooking area contains the cooking stations (e.g., oven and kettles) for cooking meat and sauce. The filling area consists of a tray-place station, filling stations, check weighing station, and a Yaguchi seamer. These stations are connected through a power conveyor. For more comprehensive description of the cooking and filling areas see TWP#14: *Revised Automation Control Strategy for Tray Pack Filling/Sealing Line.*

Presently, the casing area is not a part of the CRAMTD facility. In our simulation model, we assume the same technology for casing as in the base tray-pack lines. In the base tray-pack lines, the casing area consists of a tray washing station followed by a dryer, a videojet marker, a casing and, finally, a palletizing station. Here trays are first washed and dryed. The videojet inks the trays with all the required labeling information. The trays are then cased into batches of four. Six of these cases make up a carton.

The Simulation Model

The simulation model consists of several modules, each of which is described below. Unless otherwise stated, the input data used in these modules are stored in the experimental file.

The Scheduling Module decides, at the beginning of everyday, the product to be produced on that day. It is assumed that only one product is produced on a given day. The simulation program keeps a list of outstanding orders. The Scheduling Module scans this list and finds the order with the earliest due date. The due dates are inputs to the simulation model. The order with the earliest due date defines the next product to be produced. Once the product is selected, the available inventory is checked. Inventory of any given product may exist because, on any given day, only one product is produced and it is assumed that the production would continue till the end of the day even if the demand is already met. The difference between the actual demand and the available inventory gives the current demand for the product. If the available inventory exceeds the demand, then the order is considered to be satisfied and a new order is selected from the list of outstanding orders. Meanwhile, the inventory data is updated.

Once the product and the demand have been identified, the module routes the required raw materials or ingredients to the cooking area. The program has a record of the ingredients for each of the product types it considers. Presently, the model is set up for

four different product types, namely beef stew, beef tips with gravy, beef chunks, and mixed vegetables. This number can be increased by adding a list of ingredients that are required for any new product. This list is kept within the model itself. No major change would be needed unless the product requires more than two solid filling operations and/or more than two sauce filling operations.

The Scheduling Module also schedules and controls the start and stop of daily production. There are three staggered shifts. The cooking operation is the first to start in the morning. It is assumed that this operation starts 2 hrs prior to the start of other operations. This ensures that packaging (filling) need not be stopped for the lack of any cooked material. An estimate is made of the number of cooks (batches) required for that day and the operation comes to an end once this number is met.

The packaging operation starts at 8:00 a.m. and continues for 8 hours. At the end of the 8-hour shift, the loading of trays to the packaging line stops. The trays that are already in the system are processed all the way to the retorting operation. The retort shift starts when the first load of trays (to be defined later) becomes available. The operation continues until all the filled trays are retorted.

The next module is Inventory Control Module. This module tracks the inventory level of finished goods. The inventory level is updated at the end of each day. The trays which have passed their incubation period but have been rejected due to USDA inspection are included in the inventory records. The rejection of the trays due to USDA inspection is simulated by assigning a rejection probability to a daily production. This probability is input to the model. The trays which passed the USDA inspection are then scheduled for shipping.

There are two activities involved in the cooking operation: pre-cooking and cooling. The Cooking module simulates these two activities. The pre-cooking and cooling times are

inputs to the model.

The next module is the Packaging Module which simulates all the activities taking place in the filling and sealing stations. The packaging starts with the trays being placed on the conveyor. Each tray is modeled by a logical (SIMAN) entity. SIMAN has modeling tools to represent conveyors and physical locations (e.g., filling station or sealing station). A physical conveyor is represented by a SIMAN conveyor and physical locations by SIMAN stations. All the activities associated with a physical location can be accommodated in a SIMAN station.

The module contains SIMAN stations for various filling and sealing operations, and SIMAN conveyors connecting these various stations. The SIMAN entities move on the SIMAN conveyors from one SIMAN station to another. The physical distance between these stations is also accounted for in the simulation model.

In the model there are two solid filling stations (one for meat and one for vegetables) and two sauce filling stations. One of these sauce filling stations may be utilized for gravy pre-filling. The type of filling stations to be used depends on the product. This information is kept as a routing matrix in the experimental model.

The filling operation is assumed to be continuous and synchronized. Trays located on the conveyor move from one station to another. A proper amount of material is deposited to a tray as it passes through a filling station. The time through the station is controlled by the conveyor speed which is an input to the model. The amount of material deposited to a tray is determined according to a probability distribution. For each filling station and each product, we assume a normal distribution with particular mean and standard deviation. These parameters are input to the model. These distribution can be changed if desired.

In the actual system, once the filling is done, the tray passes through a sensor which

checks to see if any mounds are formed. If any such mounds exist the tray is diverted to another station to adjust the mound level. In the simulation model the tray is diverted through this extra operation with a given probability which is a user-input.

The next step is for the tray to go through a check-weighing station. Here, the filling weights are compared to the specifications. A tray falling outside of the specification is diverted out of the line.

Once the trays have been filled and checked, they are ready for sealing. The sealing operation is also assumed to be set to the same speed as the conveyor. After sealing the trays are loaded into racks of 72 trays each. The racks are then transported to the retorting area. This completes the Packaging Module.

The next operation is retorting, which is modeled by the Retort Module. The module takes pallets from the Packaging Module and brings them to the retorts. The number of retorts depend on the conveyor speed. This number should be selected such that the two-hour time limit between the sealing of a tray and start of a retort operation is not violated. The model checks this condition for any sealed tray and reports the violations if any. Presently, we have seven retorts in the model. This number can be changed following some minor modifications in the simulation model. The number of racks that can be accommodated in one retort cycle is dependent on the size of the retort. From the available data it is assumed to be 4 racks per retort. The retorting times are dependent on the product. This number may be changed in the simulation model if desired. The retort cycle time, which is an input to the model, includes the retorting time as well as the loading and unloading time of the retort.

Another module in the simulation model is the Casing Module. Here, the retorted trays go through washing, drying, labeling, and casing in groups of four. Then six cases are placed within a carton. The number of trays in a case and the number of cases in a

cartoon could be changed in the simulation model if desired.

The last module to explain is the Failure Module, which schedules and controls the breakdowns in the system. Any machine breakdown will cause the stoppage of the whole packaging line. Therefore, the breakdown phenomenon is simulated by sampling from a time to failure probability distribution and stopping the line operation according to the sampled time. The line remains down for a period of time, which is sampled from a repair time distribution. This whole procedure could be implemented for any machine on the line. Nevertheless, presently the model only considers the Yaguchi seamer breakdowns, because of the lack of appropriate data. The parameters of time to failure and repair time probability distributions are input to the model.

A copy of the simulation program is given in Appendix I. A copy of the source code and related files for this simulation program are stored on disk. In Section 5 we will describe the procedure for running the program.

4.Simulation of the CRAMTD Advanced MRE Pouch Line

First, we give an introduction to the CRAMTD MRE Pouch line. Then, we describe the simulation model.

The CRAMTD Advanced MRE Pouch Line

The CRAMTD Advanced MRE Pouch line is a hypothetical production line based on the technology used in the CRAMTD pilot plant and proposed for use in the CRAMTD phase II. It is composed of six major areas: the cooking area, the filling (packaging) area, an inspection area, the retort area, another inspection area, and the casing area. A schematic of this line is shown in Figure 2. The cooking area contains the cooking stations (e.g., an oven and kettles) for cooking meat and sauce. The filling area consists of a forming station, where the pouches are formed, filling stations, sealing station,

followed by three cutting stations. The forming station forms six pouches at a time into two rows of three pouches each. These stations are connected by a power conveyor. Every time that the conveyor indexes, it moves through a length of three pouches. The type and number of filling stations for the MRE Pouch line are not yet defined.

As for the sealing operation, it is performed in batches of six pouches. The cutting operation takes place in two stages. First a horizontal cutter cuts the pouches into two rows. Then two vertical cutters separate out the pouches. Following the cutting operation, all the pouches go through an inspection station. The non-defective pouches then are loaded into baskets for retorting.

The casing area is not part of the CRAMTD facility. For the simulation model, we assume a casing area which is similar to what is currently used in the industry. It includes a washing, a drying, and a video jet marker station. This area is followed by an inspection station which includes a conveyor and a number of inspectors. The inspected and accepted pouches then move into cases which are then cartoned.

The Simulation Model

Like the simulation model for traypack line, the MRE pouch model is also divided into several modules. The Scheduling module is identical to the one in traypack line model. It decides the product to be produced at the beginning of each day, routes the ingredients to the cooking area and decides on the start and stop times of the various operations. The Cooking and Inventory modules are also similar to the ones in the tray pack line model.

In the packaging area there are some noticeable differences between the two simulation models. In the tray pack model, the trays continuously move through different stations. But, in the MRE pouch model the pouches are indexed through the system. Six pouches are modeled by a single SIMAN entity as they move through the packaging line. After they come out of the cutting station, a single SIMAN entity is divided into six SIMAN

entities. Every operation in the packaging line is synchronized and executed once at the end of the each index. As in the tray pack line model, each physical station is modeled by a SIMAN station which are connected by SIMAN conveyors. The total length of these conveyors from forming to sealing stations is equivalent to fifteen indexes, that is, slightly more than the breadth of 45 pouches.

The inspection module takes sealed pouches and inspect them 100%. The model assumes that every inspected pouch has a probability of failing the inspection. Following the inspection, the pouches are palletized into batches of 1024. Two of these batch loaded in the retort simultaneously. The retorting operation is similar to that of the tray pack line.

The retorted pouches undergo a 100% inspection after they are washed and dried. This inspection station has been modeled identical to the earlier inspection station. The pouches are then cased and cartoned. The inventory module and the failure module have also been modeled as in the tray-pack line.

A copy of the simulation program is given in Appendix II. A copy of the source code and related files for this simulation program are stored on disk. The description as to how to run this program is given in Section 5.

5. Some Guidelines to Use Simulation Programs

In this section, first we describe the hardware and software requirements for running the simulation programs for the CRAMTD Advanced Tray-pack and MRE Pouch lines. Then we describe the procedure that should be followed to run these programs. Finally, we summarize the input data for these simulation programs.

Hardware and Software Requirements

- IBM PC/PS2 or IBM compatibles with minimum of 640K RAM
- Math co-processor
- DOS operating system version 3.3 or higher
- VGA/EGA monitor
- Keyboard and mouse
- SIMAN/CINEMA software (trademark of SMC)

The Procedure to Run Simulation Programs

The CRAMTD simulation disks contain the source code and the executable files for the tray-pack and MRE pouch line simulation programs. Diskette I contains files: TRAY.MOD, TRAY.EXP, TRAY.BAT, TRAYC.BAT, TRAY.LAY, TRAY.TRA, and TRAY.ENT. Diskette II contains files: MRE.MOD, MRE.EXP, MRE.BAT, MREC.BAT, MRE.LAY, MRE.TRA, and MRE.ENT. We will assume that the SIMAN software has already been installed in the computer and there exists a subdirectory called SIMAN. The steps to run the programs are as follows:

- Copy all the files in the disk to SIMAN subdirectory.
- To run the tray-pack simulation model without CINEMA animation, type TRAY <filename>, where <filename> is the file to store the output report. This report contains information defined by default in SIMAN.
- The simulation program automatically stores output data about inventory levels, daily production rates, and material usage in output files called OUTPUT.22, OUTPUT.23, OUTPUT.24. These files are in SIMAN specific format and must be processed by the SIMAN output processor prior to be readable. The procedure is as follows:
  - Type OUTPT to start the output processor. Hit "C" for color monitor.

  - At the prompt, type EXPORT:<file number>,<file number>; where the first <file number> refers to the input file (e.g., 22, 23, or 24) and the second number defines the file where the processed output will be stored.

- To run the tray-pack simulation model with CINEMA animation, type TRAYC. Once the CINEMA screen is loaded on the screen, click on the "run" button using the mouse. To stop the animation, hit ESC key and click on the "quit button using the mouse.

- To run the MRE pouch simulation program, follow the same steps by replacing "TRAY" with "MRE".

Input Data Used in Simulation Programs

Here, we summarize the input data used in the CRAMTD Advanced Tray-pack and MRE Pouch line simulation programs. The summary data is given in Table 1. In this table, PARA=PARAMETERS, SEG=SEGMENTS, ARR=ARRIVALS, TRANS=TRANSPORTER, INIT=INITIALIZE. These labels refer to headings used in the experimental files: TRAY.EXP and MRE.EXP. N/A refers to "Not Applicable". The first column in Table 1 is the input category. The second and third columns give the file names where the input data are stored for the Tray pack and MRE pouch simulations, respectively.

Table 1: Input Data Used in Simulation Programs

Input Category	Tray-Pack	MRE Pouch
Product Type	TRAY.EXP(ARRIVALS)	MRE.EXP(ARRIVALS)
Order Size	TRAY.EXP(ARRIVALS)	MRE.EXP(ARRIVALS)
Due Date	TRAY.EXP(ARRIVALS)	MRE.EXP(ARRIVALS)
Mean and S.D. for solid fill weights	TRAY.EXP(PARA)	N/A
Mean and S.D. for liquid fill weights	TRAY.EXP(PARA)	N/A
Incubation Period	TRAY.EXP(PARA)	MRE.EXP(PARA)
USDA Rejection Probability	TRAY.EXP(PARA)	MRE.EXP(PARA)
Conveyor Speed	TRAY.EXP(SEG)	TRAY.EXP(SEG)
Probability of Mound Formation	TRAY.EXP(PARA)	N/A
Specification Limits for fill Weights	TRAY.EXP(PARA)	N/A
Pallet Size	TRAY.EXP(INIT)	MRE.EXP(INIT)
Transportation Time from filling to retort	TRAY.EXP(TRANS)	MRE.EXP(TRANS)
Retort Time +Load/Unload Time	TRAY.EXP(PARA)	MRE.EXP(PARA)
Time to complete a cook	TRAY.EXP(PARA)	MRE.EXP(PARA)
Cooling Time	TRAY.EXP(PARA)	MRE.EXP(PARA)
Transportation Time from cooking to filling	TRAY.EXP(TRANS)	MRE.EXP(TRANS)
Mean Time to Failure of Yaguchi Seamer	TRAY.EXP(PARA)	N/A
Mean Time to Repair of Yaguchi Seamer	TRAY.EXP(PARA)	N/A
Maintenance Times of Fillers	TRAY.EXP(PARA)	MRE.EXP(PARA)
Time Required to Load Trays to Conveyor	TRAY.EXP(PARA)	N/A
Time Required to wipe Trays after Washing	TRAY.EXP(PARA)	N/A
Time Required for Casing	TRAY.EXP(PARA)	MRE.EXP(PARA)
Time Required for Palletizing	TRAY.EXP(PARA)	MRE.EXP(PARA)
Cook Size	TRAY.MOD	MRE.MOD
Ingredients for each product type	TRAY.MOD	MRE.MOD

References

Banks, J. and J.S. Carson, II, *Discrete Event System Simulation*, Prentice Hall, 1984.

Jafari, M.A. and V.S. Srinivasan, "Simulation Model Software Requirements Specification, Version 1.0," CRAMTD report, Rutgers University, Spet. 1990.

Pegden, C.D., *Introduction to SIMAN*, System Modeling Corp., State College, PA, 1986.

Sigethy A., T. Descovich, and T.O. Boucher, "Revised Automation Control Strategy for Tray Pack Filling/Sealing Line," Technical Working Paper (TWP) 14, CRAMTD, Rutgers University, Spet. 1990.

FIGURE 1 - PRODUCTION MODEL BASED ON CRAMTD TRAYPACK

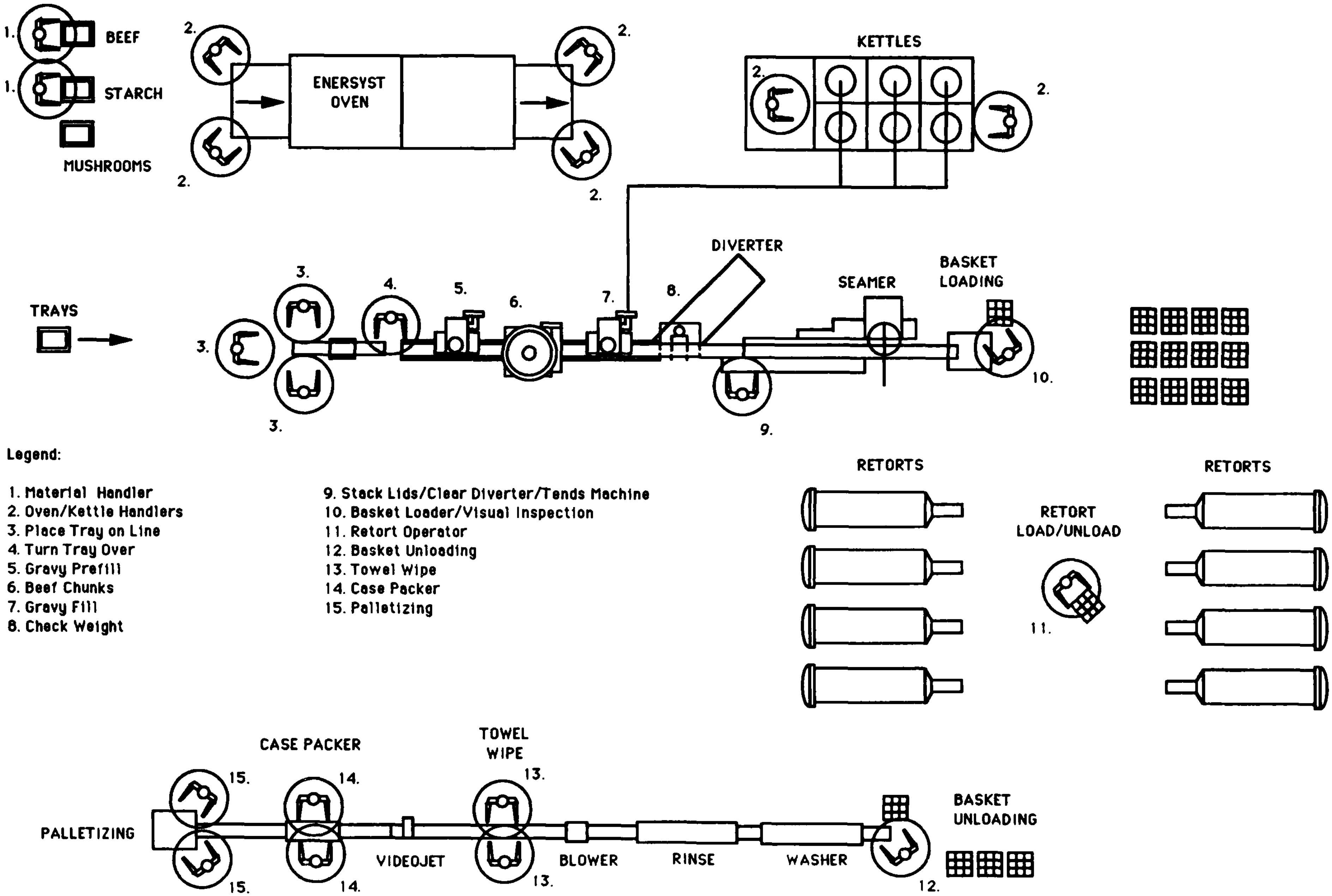
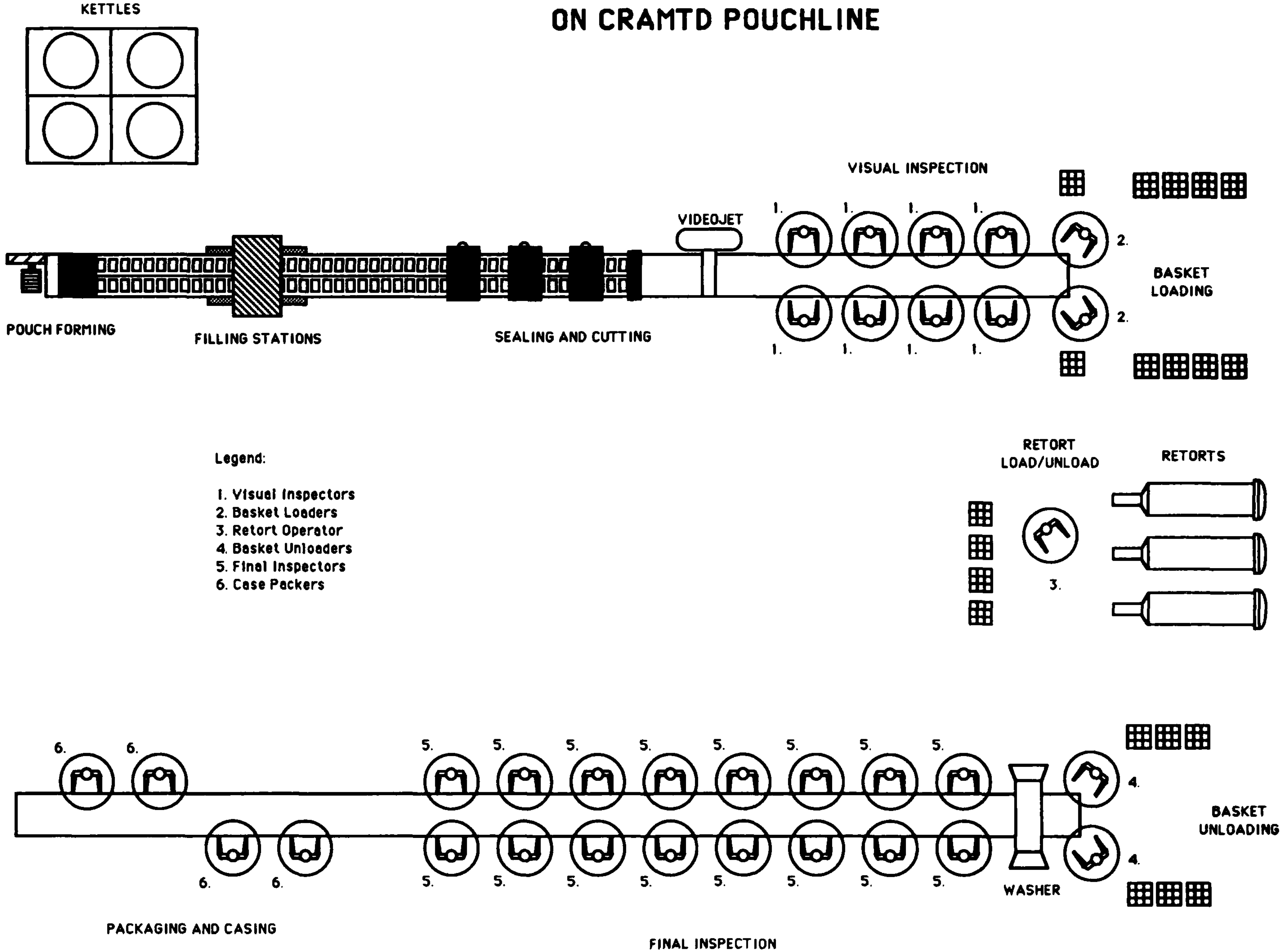


FIGURE 2 - PRODUCTION MODEL BASED ON CRAMTD POUCHLINE



APPENDIX_I
CRAMTD_TRAY_LINE_MODEL

```
BEGIN,  1,  1,yes ,TRAY    ,NO   ;
;
;-------------------------
;
SYNONYMS:
            PRODUCT = A(1):                 ! for trays
            CIN_ATTRB = A(2):               ! for trays
            PARTTYPE = A(3):                ! for trays
            DEMAND = A(2):
            DUE_DATE = A(3):
            KILL_DATE = A(4):
            INVENTORY = A(2):               ! FOR INVENTORY
            PROD_DATE = A(4):               ! FOR INVENTORY
            WT = A(4);                      ! To check weight of tray
;
;           FOR ORDERS
;           A(1)=PRODUCT
;           A(2)=DEMAND
;           A(3)=DUE DATE
;           A(4)=KILL DATE
;
;           FOR TRAYS
;           A(1)=PRODUCT
;           A(2)=CINEMA ATTRIBUTE
;           A(3)=PARTTYPE
;           A(4)=WHICH TRAY PLACE/WHICH RETORT/WEIGHT
;           A(5)=BATCH SIZE
;           A(6)=TIME IN SYSTEM
;           A(7)=TIME IN SYSTEM
;          FILLINDEX = 42
;
;
;**************************************************************************
;**************************************************************************
;
;                       SCHEDULING MODULE
;
;**************************************************************************
;**************************************************************************
;
;
;**************************************************************************
;       The input to this routine is the various orders for which the
;       the simulation is to be done. These orders are given in the
;       arrivals block of the experimental file. The parameters would
;       be the Start time of the order, The size of the order, The
;       due date and a kill date beyond which the order can be
;       assumed to be void. This block also starts/stops the machines.
;**************************************************************************
        QUEUE,80:DETACH;                 queue for oustanding orders
        CREATE,1:86400;                  create one entity every day
;**************************************************************************
;               Initailizing System parameters daily
;**************************************************************************
        ASSIGN:X(45)=X(45)+1;
        ASSIGN:D(3)=0;
        ASSIGN:D(16)=UN(74,1);
        ASSIGN:X(9)=0;
```

```
        ASSIGN:X(30)=4;
        ASSIGN:X(47)=72;
        ASSIGN:X(49)=4;
        ASSIGN:X(1)=0;
        ASSIGN:X(22)=1;
        ASSIGN:X(10)=4;
        ASSIGN:X(11)=36;
startu  BRANCH,1:
          IF,NQ(80).LT.1.AND.NQ(81).LT.1,C24:! Check for end of simulation
          ELSE,S1;
C24     ASSIGN:P(6,1)=1;
        DELAY:2*60*60:NEXT(PP9);
END2    QUEUE,83;
;        TALLY:4,NC(20);                  FOR CUMMULATIVE ORDERS 1
;        TALLY:5,NC(21);                  FOR CUMMULATIVE ORDERS 2
;        TALLY:6,NC(22);                  FOR CUMMULATIVE ORDERS 3
;        TALLY:7,NC(23);
        COUNT:13,1:DISPOSE;             ENDS SIMULATION
S1      SEARCH,80,1,NQ:(TNOW/86400).GE.'KILL_DATE'; check for cancelled orders
        BRANCH,1:
          IF, J.EQ.0 , C2:
          ELSE,        R;
R       DELAY:1;
        REMOVE: J,80,DIS:NEXT(S1);
C2      BRANCH,1:                          !check for filled existing order
          IF,NQ(81).GE.1,R1:
          ELSE,         R2;
R1      REMOVE:1,81,CONTTT:DISPOSE;             GET UNFINISHED JOB
R2      SEARCH,80,1,NQ:MIN('DUE_DATE');     FIND JOB WITH MINIMUM DUE DATE
        BRANCH,1:
          IF, J.EQ.0 , C24:
          ELSE,        RRR;
RRR     REMOVE:J,80,C3:NEXT(DIS);            GET NEW JOB
C3      ASSIGN:X(21)=X(21)+1;
        ASSIGN:A(4)=X(21);
        BRANCH,1:
          IF, 'DEMAND'.LE. P(63,'PRODUCT'),NOFIL:
          IF, P(63,'PRODUCT').GT.0,RASDE:
          ELSE,CONTTT;
NOFIL   ASSIGN:P(63,'PRODUCT')=P(63,'PRODUCT')-'DEMAND':NEXT(R2);
RASDE   ASSIGN:'DEMAND'='DEMAND'-P(63,'PRODUCT');
        ASSIGN:P(63,'PRODUCT')=0;
CONTTT  ASSIGN:X(46)=0;
        ASSIGN:X(44)=0;
        ASSIGN : P(3,1) = 'PRODUCT';         set parameter for PRODUCT
        ASSIGN : P(3,2) = 'DEMAND';          set parameter for DEMAND
        ASSIGN : P(3,3) = 'DUE_DATE';        set parameter for DUE_DATE
        ASSIGN : 'PRODUCT'='PRODUCT'+4;
        ASSIGN : P(42,1)=TF('PRODUCT',1);    filling mean for GF1
        ASSIGN : P(42,2)=TF('PRODUCT',2);    filling sigma for GF1
        ASSIGN : P(43,1)=TF('PRODUCT',3);    filling mean for VEG FILL 1
        ASSIGN : P(43,2)=TF('PRODUCT',4);    filling sigma for VEG FILL 1
        ASSIGN : P(44,1)=TF('PRODUCT',5);    filling mean for BEEF FILL
        ASSIGN : P(44,2)=TF('PRODUCT',6);    filling sigma for BEEF FILL
        ASSIGN : P(45,1)=TF('PRODUCT',7);    filling mean for GF2
        ASSIGN : P(45,2)=TF('PRODUCT',8);    filling sigma for GF2
        ASSIGN : 'PRODUCT'='PRODUCT'-4;
        BRANCH,2:
```

```
        ALWAYS,STAR:
        ALWAYS,MCCONT;
STAR    QUEUE,82;
        WAIT:1;
        ASSIGN:'DEMAND'='DEMAND'-X(46)+X(9);
        assign:x(41)=x(5)-x(41);
        BRANCH,2:
          always,inven:
          IF,'DEMAND'.LE.0,INV1:
          ELSE,Q81;
Q81     QUEUE,81:DETACH;
INV1    ASSIGN:P(63,'PRODUCT')=P(63,'PRODUCT')-'DEMAND';
DIS     DELAY:0:DISPOSE;
;
;
;************************************************************************
;                          MACHINE CONTROL
;       This block turns the machines off & on depending on the
;       signals from other blocks and  the time of day.
;       It takes into account the diffrence in time between
;       the start of the cooking operations and filling.
;************************************************************************
;
MCCONT  ASSIGN:P(6,1)=-1*P(6,1);
        BRANCH,1:
          IF,P(6,1) .EQ. -1,M2M:
          ELSE,MMM;
MMM     DELAY:2*60*60;
        START:FILCON:NEXT(PP1);
M2M     STOP:FILCON;
PP1     ALTER:TR_PL(1),P(6,1);
        ALTER:TR_PL(2),P(6,1);
        ALTER:TR_PL(3),P(6,1);
PP9     ALTER:BASK_UNL,P(6,1);
        ALTER:TOWEL_WP(1),P(6,1);
        ALTER:CASING(1),P(6,1);
        ALTER:PALLETZG(1),P(6,1);
        ALTER:TOWEL_WP(2),P(6,1);
        ALTER:CASING(2),P(6,1);
        ALTER:PALLETZG(2),P(6,1):DISPOSE;
;
;
;
;********************************************************************
;********************************************************************
;
;                          END SCHEDULING MODULE
;
;********************************************************************
;********************************************************************
;
;
;
;
;********************************************************************
;********************************************************************
;
;                          INVENTORY MODULE
```

```
;
;************************************************************************
;************************************************************************
;
;       This block keeps track of the on-hand inventory. The only input
;       is the probability that a given lot might be rejected by the
;       USDA.
;       It gives as output the on-hand inventory at any time
;**************************************************************************
;
;
INVEN     ASSIGN:D(15)=0;
          ASSIGN:'INVENTORY'=X(46)-X(9);
          ASSIGN:'PROD_DATE'=X(45)+UN(72,1);
          BRANCH,1:
            WITH,.1,USDREJ:
            ELSE,NOREJ;
USDREJ    ASSIGN:'PROD_DATE'='PROD_DATE'+20+UN(72,1);
NOREJ     ASSIGN:x(33)=x(33)+'INVENTORY';
          ASSIGN:X(27)='PROD_DATE'+10;
          BRANCH, 2 :
          ALWAYS , STOR:
          ALWAYS , RELE;
STOR      ASSIGN:D(17)=D(17)+'INVENTORY';
STOR1     QUEUE,  1:DETACH;
RELE      SEARCH,1,1,NQ:(X(45)-10).GE.'PROD_DATE';
          BRANCH ,1:
             IF,J.EQ.0, NORE:
             ELSE, CHEKM;
CHEKM     REMOVE: J,1,Q2;
          DELAY:1:NEXT(rele);         remove entity given by position J
Q2        QUEUE , 2;
          ASSIGN:D(15)=D(15)+'INVENTORY';
          ASSIGN:D(14)=D(16);
          QUEUE , 3;
          WAIT:12;
          ASSIGN:A(2)=X(22);
          BRANCH ,1:
             IF, D(15) .LT. D(16), STOR1:
             ELSE,SED14;
SED14     ASSIGN:D(16)=UN(74,1);
          BRANCH,1:
          IF, D(15) .LT. (D(16)+D(14)),ESHIP:
          ELSE,SETD;
SETD      ASSIGN:x(39)=d(16);
          ASSIGN:D(14)=D(14)+D(16):NEXT(SED14);
ESHIP     ASSIGN: D(14)=D(14)-'INVENTORY';
          ASSIGN: D(17)=D(17)-'INVENTORY';
          ASSIGN:X(22)=X(22)+1;
          BRANCH ,1:
             IF ,D(14) .GT. 0 ,DIS:
             IF ,D(14) .LE. 0,RET;
RET       ASSIGN:x(33)=D(17)-D(14);
          ASSIGN:D(16)=UN(74,1);
          ASSIGN:'INVENTORY'=-D(14):NEXT(STOR);
NORE      SIGNAL:12:DISPOSE;
;
;
```

```
;
;
;
;
;**********************************************************************
;**********************************************************************
;
;                          END INVENTORY MODULE
;
;**********************************************************************
;**********************************************************************
;
;
;
;**********************************************************************
;
;                          PCKAGING MODULE
;       The trays are filled and sealed in this module

;**********************************************************************
;
;
;
;
;
;
;
;
;**********************************************************************
;       The first station here places the trays on the conveyor.
;**********************************************************************;
;
; TRAY
         CREATE,1, 3600:24*60*60;             CREATE TRAY ONE EVERY DAY
         ASSIGN:'PARTTYPE'=     1;
         ASSIGN:'CIN_ATTRB'=    1;
         ASSIGN:'PRODUCT'= P(3,1);
         ROUTE:0,  1;                         BRANCH TO DOCK
DOCK_STN DELAY:0;
         STATION, 1;
         BRANCH,1:                           ! Check for days end
           IF,TNOW-X(45)*24*60*60.GT.36000,DIS:
           ELSE,COUNT1;
COUNT1   COUNT:     1;
         QUEUE,30;
         REQUEST,,3:RUNNER;
         TRANSPORT:RUNNER,4;
;
;**********************************************************************
;       This keeps track of the total production and helps the
;       scheduling module decide if the required production is
;        achieved.
;**********************************************************************
; STOCK
         STATION,    2;
         FREE:   RUNNER;
         BRANCH,2:
           IF,X(5).EQ.X(43).AND.NQ(46).EQ.0.AND.NQ(82).EQ.0,E:
           ALWAYS,COUNT2;
```

```
E         DELAY:1;
          BRANCH,1:
            IF,NQ(18).EQ.0.AND.NQ(47).EQ.0,END2:
            ELSE,DIS;
COUNT2    COUNT:       2: DISPOSE;
;
;
; SCRAP
          STATION,    3;
          EXIT:EXICON,16;
SCRAP     COUNT:       3: DISPOSE;
;
;
;
;**********************************************************************
;
;        The actuall filling operations start here. The first station
;        is where the empty trays are placed on the conveyor.
;        At the end of day, the production is stopped by starving this
;        station.
;
;**********************************************************************
;
;
          STATION,    4;
          FREE: RUNNER;
          ASSIGN:X(1)=X(1)+48;
          ASSIGN: 'CIN_ATTRB'= 10;            change to tray symbol
          ASSIGN: 'PRODUCT'=P(3,1);
          BRANCH,1:
            IF, (TNOW-X(45)*86400 .GE. 10*60*60),DIS:
            ELSE, TRAY_DUP;
TRAY_DUP  ASSIGN: NS=P(3,1);        pallet of tray bodies --> 48 CARTONS
          BRANCH,1:
            IF, TNOW-X(45)*86400 .GE. 10*60*60,DIS:
            ELSE, CO2;
CO2       ASSIGN:X(1)=X(1)-1;
          DUPLICATE: 21;
          BRANCH,1:
            IF, TNOW-X(45)*86400 .GE. 9.5*60*60,DIS:
            ELSE, CO3;
CO3       QUEUE,      4;
          SELECT,RAN:
          TR_PL1:
          TR_PL2:
          TR_PL3;
TR_PL1    SEIZE, 1:TR_PL(1);
          ASSIGN:A(4)=1:NEXT(TRDEL);
TR_PL2    SEIZE, 1:TR_PL(2);
          ASSIGN:A(4)=2:NEXT(TRDEL);
TR_PL3    SEIZE, 1:TR_PL(3);
          ASSIGN:A(4)=3:NEXT(TRDEL);
TRDEL     DELAY:       RN(18,1),4:MARK (6);
NON       BRANCH,2:
            ALWAYS,TURNON:
            IF, X(1)  .EQ. 2 .AND. NQ(4).EQ.10, DOCK_STN:
            IF, NQ(4) .LE. 8 .AND. X(1).GT. 0, TRAY_DUP;
TURNON    ASSIGN: X(44)=X(44)+1;      total trays produced per day
```

```
        ASSIGN: A(5)=X(44);
        ASSIGN:P(52,'PRODUCT')=P(52,'PRODUCT')+1;
        BRANCH,1:
          IF, NQ(4).EQ.0 .AND.
          NR(2)+NR(3)+NR(4) .EQ.1, LASTENT:
          ELSE, C11;
LASTENT ASSIGN: X(46)=A(5);
        ASSIGN:X(43)=X(43)+X(46);
C11     RELEASE:    TR_PL(A(4)) ;
        queue,  7;
        seize: tr_flip;
        delay: 2;
        release: tr_flip;
        COUNT:      4;
        QUEUE,  24;
        ACCESS: FILCON, 16;
        CONVEY: FILCON, 5;
;
;
;**********************************************************************
;       The next four stations are the two sauce filling stations and
;       the two solid filling stations. Between them they fill the tray,
;       with required material, while it is in motion. By changing
;       the routing it is possible to use all or any of these stations
;       in the required order.
;       The fisrt station is the Gravy/sauce prefill
;       The second is the vegetable filling station
;       The third is the solid filling station
;       The fourth is the another gravy/sauce filling station.
;**********************************************************************
;
;
;
        STATION, 5;
        QUEUE,77;
        SCAN:NQ(61).LT.3;
        EXIT:FILCON,16;
        ASSIGN:'WT'=RN(42,1);
        BRANCH,2:
          ALLWAYS,GF1:
          ALLWAYS,CON2;
GF1     ASSIGN:'CIN_ATTRB'=18;
        QUEUE,5;
        SEIZE,1:GRAVFL1;
        DELAY:2.5;
        RELEASE:GRAVFL1:DISPOSE;
CON2    QUEUE,61;
        ACCESS:CON2,16;
        CONVEY:CON2,6;
;
;
;
;
        STATION,6;
        EXIT:CON2,16;
CON3    BRANCH,1:
          IF,'PRODUCT'.EQ.4,NOMEAT:
          ELSE,MEAT;
```

```
MEAT      QUEUE,67;
          ACCESS:CON3,16;
          CONVEY:CON3,7;
NOMEAT    ASSIGN:'WT'='WT'+RN(43,1);
          BRANCH,2:
           ALLWAYS,Q68:
           ALLWAYS,VEGFIL;
VEGFIL    QUEUE,11;
          SEIZE,1:VEGFIL;
          DELAY:2.5;
          RELEASE:VEGFIL:DISPOSE;
Q68       QUEUE,68;
          ACCESS:CON4,16;
          CONVEY:CON4,8;
;
;
;
;
          STATION,7;
          EXIT:CON3,16;
          BRANCH,1:
            WITH,0.05,CDATA:
            WITH,0.95,NDATA;
CDATA     ASSIGN:X(19)=RN(44,1);
          ASSIGN:'WT'='WT'+X(19):NEXT(BEEFBR);
NDATA     ASSIGN:'WT'='WT'+RN(44,1);
BEEFBR    BRANCH,2:
            ALLWAYS,BEEF:
            ALLWAYS,CON5;
BEEF      ASSIGN:'CIN_ATTRB'=18;
          QUEUE,6;
          SEIZE,1:BEEF;
          DELAY:2.5;
          RELEASE:BEEF:DISPOSE;
CON5      QUEUE,8;
          ACCESS:CON5,16;
          CONVEY:CON5,8;
;
;
;
;

          STATION,8;
;          BRANCH,1:
;            IF,'PRODUCT' .EQ. 4, EXCON4:
;            ELSE, EXCON5;
;EXCON4    EXIT:CON4,16:NEXT(BRANCH);
EXCON5    EXIT:CON5,16;
          ASSIGN:'WT'='WT'+P(45,1);
BRANCH    BRANCH,2:
            ALLWAYS,GF2:
            ALLWAYS,CON8;
GF2       ASSIGN:'CIN_ATTRB'=18;
          QUEUE,9;
          SEIZE,1:GRAVYFL2;
          DELAY:2.5;
          RELEASE:GRAVYFL2:DISPOSE;
CON8      QUEUE,66;
```

```
          ACCESS:CON8,16;
          CONVEY:CON8,22;
;
;
;*********************************************************************
;
;       After the tray has been filled the mounds have to be
;       crushed to avoid the possiblity of bad seal.
;
;*********************************************************************
;
;

          STATION,    22;
          EXIT:  CON8,16;
          BRANCH,  2:
            ALWAYS, MCRUSH:
            ALWAYS, CON6;
MCRUSH    QUEUE, 48;
          SEIZE,1: M_CRUSHER;
          DELAY:2.5;
          RELEASE: M_CRUSHER:DISPOSE;
CON6      QUEUE, 49;
          ACCESS:CON6,16;
          CONVEY:CON6,9;
;
;
;
;*********************************************************************
;       The tray at this point is checked for its weight.
;       If Specs provided decide is the tray should be rejected or
;       not.
;*********************************************************************
;
;
          STATION,    9;
          EXIT:  CON6,16;
          ASSIGN:'PRODUCT'='PRODUCT'+7;
          TALLY:'PRODUCT',A(8);
          ASSIGN:'PRODUCT'='PRODUCT'-7;
;         BRANCH,1:
;            IF,A(5).EQ.X(46),RELAST:
;           IF,P(46,'PRODUCT') .GT. 'WT', REJECT:
;            ELSE,PRREJ;
;RELAST    ASSIGN:X(43)=X(43):NEXT(COUNT11);
;REJECT    ASSIGN:X(9)=X(9)+1;
;          QUEUE,10;
;          ACCESS:EXICON,16;
;          CONVEY:EXICON,3;
;PRREJ     BRANCH,1:
;
;            WITH,P(47,'PRODUCT'), REJECT:
;            ELSE, COUNT11;
;COUNT11   COUNT:      5;
          QUEUE,62;
          ACCESS:CON7,16;
          CONVEY:CON7,12;
;
```

```
;
;
;*******************************************************************
;*******************************************************************
;
;                         END PACKAGING MODULE
;
;*******************************************************************
;*******************************************************************
;
;
;
;
;*******************************************************************
;*******************************************************************
;
;                         RETORTING MODULE
;
;*******************************************************************
;*******************************************************************

;*******************************************************************
;       At this point the trays are sealed and loaded into
;       pallets for loading into the retort.
;*******************************************************************
;
         STATION,     12;
         EXIT:CON7,16;
         QUEUE,       12;
         SEIZE, 1:    BASK_LDR ;
         DELAY:       RN( 26,2 ),12;
         RELEASE:     BASK_LDR ;
         TALLY:       'PRODUCT',INT(6);
         COUNT:       6;
         BRANCH,1:
           IF, A(5).EQ.X(46), ALTCOM:
           ELSE, COM;
ALTCOM   SIGNAL:1;
         BRANCH,2:
           ALWAYS,CCC:
           ALWAYS,MCCONT;
CCC      ASSIGN:X(47)=NQ(26)+1:NEXT(COM);
COM      QUEUE,       26;
         COMBINE: X(47), LAST;
         ASSIGN:      'CIN_ATTRB'=  8;          change to basket symbol
         ASSIGN:      'PARTTYPE' = 1;
         QUEUE,       27;
         REQUEST,,32: RET_LDR;
         TRANSPORT : RET_LDR,13;
         STATION,     13;
         FREE:        RET_LDR;
         BRANCH,1:
           IF,A(5).EQ.X(46),LAST:
           ELSE,CCCC;
LAST     ASSIGN:A(5)=X(47);
         ASSIGN:X(47)=72;
         ASSIGN:X(30)=NQ(43)+1:NEXT(Q43);
CCCC     ASSIGN:A(5)=72:NEXT(Q43);
```

```
Q43       QUEUE,       43;
          GROUP:X(30),LAST;
          QUEUE,       13;
          SELECT,RAN:
           RETORT1:
           RETORT2:
           RETORT3:
           RETORT4:
           RETORT5:
           RETORT6:
           RETORT7;
RETORT1   SEIZE, 1:    RETORT(1);
          ASSIGN:A(4)=1:NEXT(RETREL);
RETORT2   SEIZE, 1:    RETORT(2);
          ASSIGN:A(4)=2:NEXT(RETREL);
RETORT3   SEIZE, 1:    RETORT(3);
          ASSIGN:A(4)=3:NEXT(RETREL);
RETORT4   SEIZE, 1:    RETORT(4);
          ASSIGN:A(4)=4:NEXT(RETREL);
RETORT5   SEIZE, 1:    RETORT(5);
          ASSIGN:A(4)=5:NEXT(RETREL);
RETORT6   SEIZE, 1:    RETORT(6);
          ASSIGN:A(4)=6:NEXT(RETREL);
RETORT7   SEIZE, 1:    RETORT(7);
          ASSIGN:A(4)=7:NEXT(RETREL);
RETREL    ASSIGN:      'CIN_ATTRB'=  9;        change to 4 basket symbol
          DELAY:       120*60,13;
          RELEASE:     RETORT(A(4)) ;
          COUNT:       7;
          SPLIT;
          ASSIGN:      'CIN_ATTRB'=  8;      change to basket symbol
          QUEUE,       28;
          REQUEST,,33: RET_LDR;
          TRANSPORT:RET_LDR,14;
;
;
;
;
;
;*********************************************************************
;*********************************************************************
;
;                          END RETORTING MODULE
;
;*********************************************************************
;*********************************************************************
;
;
;
;
;*********************************************************************
;*********************************************************************
;
;                          CASING MODULE
;
;*********************************************************************
;*********************************************************************
;
```

```
;
;
;**********************************************************************
;        The pallets are removed from the retort and sent over to this
;        module. Here the trays are removed from the pallets and cased.
;        The various operations involved are washing, drying, labeling
;        and cartooning;
;**********************************************************************
;
;
;**********************************************************************
;        This station unloads the trays from the pallet and places it
;        on the conveyor where it is washed.
;**********************************************************************

          STATION,     14;
STN14     FREE:        RET_LDR;
          BRANCH,1:
            IF, NQ(14) .EQ. 0,DUPA5:
            ELSE,Q72;
Q72       QUEUE, 72;
          WAIT:2,1;
DUPA5     DUPLICATE:A(5)-1;
          ASSIGN:     'CIN_ATTRB'= 10: MARK (6);       change to tray symbol
          QUEUE,       14;
          SEIZE, 1:    BASK_UNL ;
          BRANCH,1:
            IF, NQ(14) .EQ. 0, SIG2:
            ELSE,CONTIU2;
SIG2      SIGNAL:2;
CONTIU2   DELAY:       RN(28,4),14;
          RELEASE:     BASK_UNL ;
          COUNT:       8;
          QUEUE,       44;
          ACCESS: WASH_DRY, 16;
          CONVEY: WASH_DRY, 15;
;
;
;
;**********************************************************************
;        Once the trays have been washed they have to dryed clean.
;        This operation is modeled in the this station.
;**********************************************************************
;
;
          STATION,     15;
          EXIT:   WASH_DRY, 16;
          QUEUE,       15;
          SELECT, RAN:
          TW1:
          TW2;
TW1       SEIZE, 1:    TOWEL_WP(1) ;
          ASSIGN:A(4)=1:NEXT(TWDEL);
TW2       SEIZE, 1:    TOWEL_WP(2) ;
          ASSIGN:A(4)=2;
TWDEL     DELAY:       RN(29,5),15;
          RELEASE:     TOWEL_WP(A(4)) ;
          COUNT:       9;
```

```
        ROUTE: P(10,7), 16;
;
;
;***********************************************************************
;       The trays are then labled and cartooned in groups
;       four. Six of these cartoons are then put into a cardboard box.
;       These operatins have been modeled in the following few blocks.
;***********************************************************************
        STATION,    16;
        COUNT:      10;
        ASSIGN:'PRODUCT'='PRODUCT'+13;
        COUNT:'PRODUCT';
        ASSIGN:'PRODUCT'='PRODUCT'-13;
        ROUTE: P(10,8), 17;
;
;
        STATION,    17;
        QUEUE,      17;
        SELECT ,RAN:
        CAS1:
        CAS2;
CAS1    SEIZE, 1:   CASING(1) ;
        ASSIGN:A(4)=1:NEXT(CASDEL);
CAS2    SEIZE, 1:   CASING(2) ;
        ASSIGN:A(4)=2:NEXT(CASDEL);
CASDEL  ASSIGN:     'CIN_ATTRB' = 12;      change to case symbol
        DELAY:      RN(31,7),17;
        RELEASE:    CASING(A(4)) ;
        ASSIGN:X(37)=X(37)+1;
        ASSIGN:X(6)=X(6)+1;
        BRANCH,1:
          IF,X(43).EQ.X(6).AND.NQ(82).EQ.0,N3:
          IF,A(1).EQ.P(8,1).OR.NQ(45).EQ.0,N1:
          ELSE,N2;
N3      ASSIGN:X(10)=NQ(45)+1:NEXT(N1);
N2      ASSIGN:X(10)=NQ(45);
        REMOVE:NQ,45,Q45;
        DELAY:1;
N1      ASSIGN:P(8,1)=A(1);
Q45     QUEUE,      45;
        COMBINE:  X(10),LAST;
        ASSIGN:A(2)=X(10);
        ASSIGN:X(10)=4;
        ASSIGN:     'PARTTYPE' = 1;
        COUNT:      11;
        ROUTE: P(10,9), 18;
;
;
        STATION,    18;
        ASSIGN:X(5)=X(5)+A(2);
        QUEUE,      18;
        SELECT ,RAN:
        PALL1:
        PALL2;
PALL1   SEIZE, 1:   PALLETZG(1);
        ASSIGN:A(4)=1:NEXT(PADEL);
PALL2   SEIZE, 1:   PALLETZG(2);
        ASSIGN:A(4)=2:NEXT(PADEL);
```

```
PADEL    ASSIGN:      'CIN_ATTRB' = 13;        change to pallet symbol
         DELAY:       RN(32,8),18;
         RELEASE:     PALLETZG(A(4)) ;
         BRANCH,1:
           IF,X(5).EQ.X(43).AND.NQ(82).EQ.0,NN3:
           IF,A(1).EQ.P(8,3).OR.NQ(46).EQ.0,NN1:
           ELSE,NN2;
NN3      ASSIGN:X(11)=NQ(46)+1:NEXT(NN1);
NN2      ASSIGN:X(11)=NQ(46);
         REMOVE:NQ,46,Q46;
         DELAY:1;
NN1      ASSIGN:P(8,3)=A(1);
Q46      QUEUE,       46;
         COMBINE:   X(11),LAST;
         ASSIGN:X(11)=36;
OP       COUNT:       12;
         QUEUE,       47;
         REQUEST,,30:RUNNER,      3.0000;
         TRANSPORT:  RUNNER,2;
;
;*********************************************************************
;*********************************************************************
;
;                        END CASING MODULE
;
;*********************************************************************
;*********************************************************************
;
;
;*********************************************************************
;*********************************************************************
;
;                        COOKING MODULE
;
;
;*********************************************************************
;*********************************************************************
;
;
;*********************************************************************
;       This is the first operation to start on any given day.
;       It can be divided into two activites Cooking/Cooling
;*********************************************************************
;
;
         STATION,     19;
         FREE:        RUNNER;
         BRANCH,1:
           IF, X(48) .LT. 4, X49:
           ELSE, BEEF_DUP;
X49      ASSIGN:X(49)=X(48);
BEEF_DUP DUPLICATE:  X(49);        2000 # beef pallet --> 4 beef kettle batches
         ASSIGN:X(48)=X(48)-1;
         QUEUE,       19;
         SEIZE, 1:   BEEF_KET ;
         DELAY:       RN(33,9),19;
         BRANCH,2:
           ALWAYS,CONREL:
```

```
          IF, NQ(19) .EQ. 2 .AND. X(48).GE.0, DOCK_STN;
CONREL   RELEASE:     BEEF_KET ;
         BRANCH,      2:
           ALWAYS,    BEEFPART:
           ALWAYS,    BROTHPRT;
BEEFPART ASSIGN:'CIN_ATTRB'=11;       change to meat tote symbol
         QUEUE,       20;
         REQUEST,,31: K_ASSIST;
         TRANSPORT : K_ASSIST, 20;
;
;
         STATION,     20;
         BRANCH,1:
           IF,'PRODUCT'.NE.4,FR1:
           ELSE,FR2;
FR1      FREE:        K_ASSIST;
         ASSIGN:      X(12) = X(12) + P(4,2):DISPOSE ;
FR2      FREE:        RUNNER;
         ASSIGN:D(1)=0;
         ASSIGN:      X(49) = X(49) + P(4,2):DISPOSE ;
;
;
BROTHPRT DELAY:0;
         STATION,     21;
         QUEUE,       21;
         SEIZE, 1:    GRAVY_K ;
         ASSIGN: X(14) = X(14) - 1;          decrement starch supply counter
         ASSIGN: X(13) = X(13) - 1;          decrement peas supply counter
         BRANCH,1:
           IF, X(13) .LE. P(5,3),SIG3:
           ELSE,CONTIU3;
SIG3     SIGNAL:3;
CONTIU3  BRANCH,1:
           IF, X(14) .LE. P(5,4),SIG4:
           ELSE,CONTIU4;
SIG4     SIGNAL:4;
CONTIU4  DELAY:       RN(35,2);
         RELEASE:     GRAVY_K ;
         ASSIGN:      X(18) = X(18) + P(4,8):DISPOSE;
;
;
;*********************************************************************
;*********************************************************************
;
;                         END COOKING MODULE
;
;*********************************************************************
;*********************************************************************
;
;
;
;*********************************************************************
;*********************************************************************
;
;                         FAILURE MODULE
;
;*********************************************************************
;*********************************************************************
```

```
;
;
;*********************************************************************
;       This module schedules the failures and when it occurs stops
;       system. It also decides the repair times by sampling a
;       distribution. After this period of time it starts the system.
;*********************************************************************
;
;
FAIL     ALTER:TR_PL(1),-1;
         ALTER:TR_PL(2),-1;
         ALTER:TR_PL(3),-1;
         ALTER:BASK_UNL,-1;
         STOP:FILCON;
         STOP:CON2;
         STOP:CON3;
         STOP:CON4;
         STOP:CON5;
         STOP:CON6;
         STOP:EXICON;
         STOP:CON7;
         STOP:CON8;
         DELAY:D(40);
         START:FILCON;
         START:CON2;
         START:CON3;
         START:CON4;
         START:CON5;
         START:CON6;
         START:EXICON;
         START:CON7;
         START:CON8;
         ALTER:TR_PL(1),+1;
         ALTER:TR_PL(2),+1;
         ALTER:TR_PL(3),+1;
         ALTER:BASK_UNL,+1;
         DELAY:0:DISPOSE;

         CREATE,1,EX(65,1):EX(65,1);     YAGUCHI
         BRANCH,1:
           IF,TNOW .gt. X(45)*86400+34200
            .OR. MR(2).EQ.0, DIS:
           ELSE, MVEG;
MVEG     ASSIGN:D(40)=EX(69,1);
         BRANCH,1:
           IF, TNOW+D(40) .GT. X(45)*86400+34200,
           SETD401:
           ELSE, FAIL;
SETD401   ASSIGN:D(40)=X(45)*86400+34200 - TNOW:NEXT(FAIL);
END;
;
;
;*********************************************************************
;*********************************************************************
;
;                         END FAILURE MODULE
;
;*********************************************************************
```

```
;***********************************************************************
;
;
;
;***********************************************************************
;***********************************************************************
;
;                        END OF THE MODEL FILE
;
;***********************************************************************
;***********************************************************************;
;
;
;
;
;
;***********************************************************************
;***********************************************************************

                          EXPERIMENTAL FILE

;***********************************************************************
;***********************************************************************
;
;
;***********************************************************************
;***********************************************************************
;
;
;***********************************************************************
;       This file contains  the data required to run the model.
;       Some of this data is user input but a good part of it
;       is the data required by siman. It also has the definition
;       of the various stations, resources etc used in the model.
;***********************************************************************
;
BEGIN,  1,  1,YES,NO  ;
PROJECT, TRAY PACK PRODUCTION,XXXXXXX              , 5/15/1990  ;
;
DISCRETE,  700,   8,  90,  35,   2;
;
;***********************************************************************
;       The block intialises certain varibles like pallet size
;***********************************************************************
INITIALIZE , X(47)=72,     ! Palet size
             X(45)=-1;
;
;
;
;***********************************************************************
;       The arrivals block stores the various orders that have to
;       be processed.
;       1(order #),QUEUE(80)(queue #),0(Time when orders is to be created)
;       1(batch size of the orders),2( product type),21856(order size)
;       ,1(due date, in days),9999(kill date , in days):
;***********************************************************************
;
;
```

```
ARRIVALS: 1,QUEUE(80),0,1,2,21856,1,9999:
          2,QUEUE(80),0,1,2,20000,2,9999:
          3,QUEUE(80),0,1,2,21856,1,9999:
          4,QUEUE(80),0,1,2,20000,2,9999:
          5,QUEUE(80),0,1,2,21856,1,9999:
          6,QUEUE(80),0,1,2,20000,2,9999:
          7,QUEUE(80),0,1,2,21856,1,9999:
          8,QUEUE(80),0,1,2,20000,2,9999:
          9,QUEUE(80),0,1,2,21856,1,9999:
         10,QUEUE(80),0,1,2,20000,2,9999;
;         3,QUEUE(80),0,1,3,1000,5,9999:
;         4,QUEUE(80),0,1,4,1500,7,9999;
;         5,QUEUE(80),0,1,1,3000,12,9999:
;         6,QUEUE(80),0,1,3,1000,13,9999:
;         7,QUEUE(80),0,1,2,3500,14,9999:
;         8,QUEUE(80),0,1,4,6000,14,9999;
;
;
;
;*********************************************************************
;       The  following decide the information that is to be put in
;       the siman output report. Most of this information is to
;       ensure that the model went through its noraml execution.
;*********************************************************************
;
;
TALLIES      :1,SYS_TIME_P1:
              2,SYS_TIME_P2:
              3,SYS_TIME_P3:
              4,SYS_TIME_P4:
              5,BEEF_PRODUCT_1:
              6,BEEF_PRODUCT_2,2:
              7,BEEF_PRODUCT_3,3:
              8,VEG_PRODUCT_1:
              9,VEG_PRODUCT_2:
             10,VEG_PRODUCT_3:
             11,VEG_PRODUCT_4;
;
  COUNTERS:
              1,DOCK:
              2,STOK:
              3,SCRAP:
              4,TR_PLACE:
              5,FCW:
              6,BL:
              7,RET:
              8,BUNL:
              9,TW:
             10,VJ:
             11,CAS:
             12,PAL:
             13,end,1:
             14,PRODUCT 1:
             15,PRODUCT 2:
             16,PRODUCT 3:
             17,PRODUCT 4;
;
;*********************************************************************
```

```
;       This gives a list of resources used in the model.
;*********************************************************************
RESOURCES:    1,DOCK    ,     0:
            2-4,TR_PL,      0,0,0:
              5,TR_FLIP ,         1:
              6,MATL2,            0:
              7,beef,            1:
              8,vegfil,            1:
              9,GRAVFL1,         1:
             10,CHKWEIGH,         1:
             11,FINAL_CW,         1:
             12,BASK_LDR,         1:
             13,RETORTS ,         0:
             14,BASK_UNL,         0:
          15-16,TOWEL_WP,       0,0:
          17-18,CASING  ,       0,0:
             19,BEEF_KET,         0:
             20,MT_DOCK ,         0:
             21,GRAVY_K ,         0:
             22,MATL4,            0:
             23,MATL5,            0:
             24,GRAVYFL2,         1:
             25-31,RETORT,   1, 1, 1, 1, 1, 1, 1:
             32-33,PALLETZG,       0,0:
             34,M_CRUSHER         1;
;
;*********************************************************************
;       This section gives the parameters for the various activities.
;       A lot of parameters here can be called SIMAN overhead since
;       they are not involved with the system understudy.
;       The Mean and SD for the filling activities, incubation period,
;       Cook times, cook size, cooling times, failure times etc are
;       given here.
;       The filling parameters are parameter # 43-48
;       The first parameter is the mean and the second the SD.
;       The incubation period is parameter # 73
;       The weight specifications can be given in parameter #52,53
;       The retort cycle time is parameter # 27
;       The cooking and cooling times are parameter #33 & #35
;       The mean time to failure of the yaguchi seamer #68
;       The mean time to repair of the yaguchi seamer #69
;       The time to place trays on conveyor #18
;       Time required to wipe trays #29
;       Time required for casing #30
;       Time required for paletizing #32
;*********************************************************************
PARAMETERS:
1,      !  MTTF
       10000000.0,    !  gf1
       10000000.0,    ! vegfil
       10000000.0,    !  gf2
       10000000.0,    !  beef
       1000000.0,    !        5,TR_FLIP
           300.0,    !        6,POTATOES
       1000000.0,    !        7,MEAT_FIL
       1000000.0,    !        8,CARROTS
       1000000.0,    !        9,GRAVYFL1
       1000000.0,    !       10,CHKWEIGH
```

```
          1000000.0,    !         11,FINAL_CW
          1000000.0,    !         12,BASK_LDR
          1000000.0,    !         13,RETORTS
          1000000.0,    !         14,BASK_UNL
          1000000.0,    !         15,TOWEL_WP
          1000000.0,    !         16,VIDEOJET
          1000000.0,    !         17,CASING
          1000000.0,    !         18,PALLETZG
          1000000.0,    !         19,BEEF_KET
          1000000.0,    !         20,MT_DOCK
          1000000.0,    !         21,GRAVY_K
          1000000.0,    !         22,MAT4
          1000000.0,    !         23,MAT5
          1000000.0,    !         24,MAT6
          1000000.0:    !         25,GRAVYFL2
 2,       !   MTTR
          900.0,    !         GF1
          900.0,    !     VEG
          900.0,    !  BEEF
          900.0,    !   GF2
          000000.0,    !          5,TR_FLIP
              30.0,    !          6,POTATOES
          000000.0,    !          7,MEAT_FIL
          000000.0,    !          8,CARROTS
          000000.0,    !          9,GRAVYFIL
          000000.0,    !         10,CHKWEIGH
          000000.0,    !         11,FINAL_CW
          000000.0,    !         12,BASK_LDR
          000000.0,    !         13,RETORTS
          000000.0,    !         14,BASK_UNL
          000000.0,    !         15,TOWEL_WP
          000000.0,    !         16,VIDEOJET
          000000.0,    !         17,CASING
          000000.0,    !         18,PALLETZG
          000000.0,    !         19,BEEF_KET
          000000.0,    !         20,MT_DOCK
          000000.0,    !         21,GRAVY_K
          000000.0,    !         22,MAT4
          000000.0,    !         23,MAT5
          000000.0,    !         24,MAT6
          000000.0:    !         25,GRAVYFL2
   3, 0,
      0,
      0:
   4, 0,    ! 1                                 ***  INCREMENTAL SUPPLY  ***
      190,  ! 2 BEEF     TRAY PORTIONS / BEEF KETTLE BATCH
        8,  ! 3 PEAS     GRAVY KETTLE BATCHES / PALLET
       38,  ! 4 STARCH    GRAVY KETTLE BATCHES / PALLET
      926,  ! 5 POTATOES  TRAY PORTIONS / PALLET
      909,  ! 6 CARROTS   TRAY PORTIONS / PALLET
     4800,  ! 7 LIDS      TRAY LIDS / PALLET
      211:  ! 8 GRAVY     TRAY PORTIONS / GRAVY KETTLE BATCH
    5,    4,  !  TRAYS    FROM PALLET           ***  REORDER COUNTER VALUES  ***
          1,  !  BEEF     BEEF KETTLE PORTIONS
          1,  !  PEAS     GRAVY KETTLE PORTIONS
          2,  !  STARCH   GRAVY KETTLE PORTIONS
        120,  !  POTATOES TRAY PORTIONS
        120,  !  CARROTS  TRAY PORTIONS
```

```
        240, !  LIDS     TRAY LIDS
          2, !  TRAYS    FROM CARTON
          1: !  TRAYS    FROM RETORTS
     6,-1:
     7, 0:
     8, 0,
        0,
        0,
        0:
     9, 0:
    10, 5, !  1   TR_FLIP  TO GRAVYFL1        *****  ROUTE TIMES  *****
        5, !  2   GRAVYFL1 TO MAT1
        5, !  3   MAT4     TO MAT2
        5, !  4   MAT5     TO LIQ2
        5, !  5   LIQ2     TO CHKWEIGH
        5, !  6   FINAL_CW TO BASK_LDR
        5, !  7   TOWEL_WP TO VIDEOJET
        5, !  8   VIDEOJET TO CASING
        5, !  9   CASING   TO PALLETZG
        5, !  10  PALLETZG TO DOCK
        5, !  11  BK_DOCK  TO BEEF_K
        5, !  12  MT_DOCK  TO DOCK
        5, !  13  GK_DOCK  TO GRAVY_K
        5, !  14  PEA_DOCK TO DOCK
        5, !  15  STR_DOCK TO DOCK
        5, !  16  PO_DOCK  TO DOCK
        5, !  17  CA_DOCK  TO DOCK
        5, !  18  LD_DOCK  TO DOCK
        5, !  19  MATL4_D  TO DOCK
        5, !  20           TO DOCK
        5: !  21  MATL5_D  TO DOCK
    11, 0, !  1   DOCK                   *****  STATION DELAY TIMES  *****
        0, !  2   STOCK
        0, !  3   SCRAP
       12, !  4   TR_PLACE
        4, !  5   TR_FLIP
        4, !  6   POTATOES
        4, !  7   MEAT_FIL
        4, !  8   CARROTS
        4, !  9   GRAVYFIL
        4, !  10  CHKWEIGH
        4, !  11  FINAL_CW
        4, !  12  BASK_LDR
     7200, !  13  RETORTS
        4, !  14  BASK_UNL
        4, !  15  TOWEL_WP
        4, !  16  VIDEOJET
        4, !  17  CASING
        4, !  18  PALLETZG
     1800, !  19  BEEF_KET
        4, !  20  MT_DOCK
     1800, !  21  GRAVY_K
        4, !  22  GF_DOCK
        4, !  23  PEA_DOCK
        4, !  24  STR_DOCK
        4, !  25  PO_DOCK
        4, !  26  CA_DOCK
        4: !  27  LD_DOCK
```

```
12, 0, !  1  ************************  BUFFER LIMITS *************************
     4, !  2  maximum seamer queue
   844, !  3  maximum X(18), gravy supply counter
     4, !  4  maximum tray_fl queue
     4, !  5  maximum tray_pl queue
     4, !  6  maximum mat4 queue
     4, !  7  maximum mat1 fill queue
     4, !  8  maximum mat5 queue
     4, !  9  maximum gravy fill 1 queue
     4, !  10 maximum checkweigh queue
     4: !  11 maximum final_cw queue
13,0:
14,0:
15,     0, 0: !  1   DOCK           *****   STATION DELAY TIMES   *****
16,     0, 0: !  2   STOCK
17,     0, 0: !  3   SCRAP
18,   11.25, 0: !  4   TR_PLACE
19,     15, 1: !  5   TR_FLIP
20,     15, 1: !  6   POTATOES
21,     15, 1: !  7   MEAT_FIL
22,     15, 1: !  8   CARROTS
23,     15, 1: !  9   GRAVYFIL
24,     3, 0: !  10  CHKWEIGH
25,     3, 0: !  11  FINAL_CW
26,     3, 0: !  12  BASK_LDR
27, 7200, 1: !  13  RETORTS
28,     3, 0: !  14  BASK_UNL
29,     3, 0: !  15  TOWEL_WP
30,     3, 0: !  16  VIDEOJET
31,     3, 0: !  17  CASING
32,     3, 0: !  18  PALLETZG
33, 1800, 1: !  19  BEEF_KET
34,     15, 1: !  20  MT_DOCK
35, 1800, 1: !  21  GRAVY_K
36,     15, 1: !  22  GF_DOCK
37,     15, 1: !  23  PEA_DOCK
38,     15, 1: !  24  STR_DOCK
39,     15, 1: !  25  PO_DOCK
40,     15, 1: !  26  CA_DOCK
41,     15, 1: !  27  LD_DOCK
42,     5, 1: !  GF1               **** FILLING AMOUNT PARAMETERS****
43,     5, 1: !  VEG FILL 1
44,     82.0, .39: !  BEEF FILL
45,     5, 1: !  GF2
46,   0,0, 0, 0: ! MIN WT FOR TRAYS PRODUCT 1 TO 4
47,     0,0,0,0: !  PROBABLISTIC REJECT FOR THE VARIOUS PROD
48,     5, 1: !  FILLING STATION # 7
49,     5, 1: !  GRAVYFIL 2 DELAY
50,     5, 1: !  MATL4 DELAY
51,     15, 1: ! MATL5 DELAY
52,     2000,2000:!wt specs
53,     2000,2000:!wt specs
54,     0,0,0,0:!MATL4
55,     0,0,0,0:!MATL5
56,     0,0,0,0:!MATL2
57,     0,0,0,0:!MATL3
58,     0,0,0,0:!GF2
59,     0,0,0,0:!
```

```
 60,     0,0,0,0:!
 61,     0,0,0,0:!
 62,     0,0,0,0:!
 63,     0,0,0,0:!INVENTORY
 64,     10000.0: !**************FAILURE TIMES GF1
 65,     10800.0: !**************FAILURE TIMES VEG
 66,     10000.0: !**************FAILURE TIMES MEAT
 67,     10000.0: !**************FAILURE TIMES GF2
 68,     900.0 :
 69,     600.0 :
 70,     900.0:
 71,     900.0:
 72,     0,4:
 73,     10: ! Incubation period
 74,     3840,4608;
;*************************************************************************
;       The speed of the transporter is given in this block
;*************************************************************************
transporters:   !    CAP  DIS SET #      VELOCITY   INITIAL STATN & STATUS
    1,RUNNER  ,      1,     1,           3.0000,          1-A:
    2,K_ASSIST,      1,     2,           3.0000,         20-A:
    3,RET_LDR ,      1,     3,           3.0000,         12-A;
distances:
1,   1-32,     ! RUNNER TRANSPORT                                       from
     10,10,50,0,0,0,0,0,0,0,0,0,0,0,0,0,50,40,0,0,0,60,60,50,55,65,
                                             50,60,50,50,50/! DOCK
        5, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,50, 0,0,0,0, 0, 0, 0, 0, 0,
                                             0, 0, 0, 0, 0/! STOCK
           0,0,0,0,0,0,0,0,0,0,0,0,0,0,50, 0,0,0,0, 0, 0, 0, 0, 0,
                                             0, 0, 0, 0, 0/! SCRAP
             0,0,0,0,0,0,0,0,0,0,0,0,0,30, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! TR_PLAC
               0,0,0,0,0,0,0,0,0,0,0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! TR_FLIP
                 0,0,0,0,0,0,0,0,0,0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! POTATOE
                   0,0,0,0,0,0,0,0,0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! MEAT_FI
                     0,0,0,0,0,0,0,0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! CARROTS
                       0,0,0,0,0,0,0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! GRAVYFI
                         0,0,0,0,0,0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! CHKWEIG
                           0,0,0,0,0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! FINAL_C
                             0,0,0,0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! BASK-LD
                               0,0,0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! RETORTS
                                 0,0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! BASK_UN
                                   0,0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! TOWEL_W
                                     0, 0, 0,0,0,0, 0, 0, 0, 0, 0,
                                           0, 0, 0, 0, 0/! VIDEOJE
                                        0, 0,0,0,0, 0, 0, 0, 0, 0,
                                            0, 0, 0, 0, 0/! CASING
```

```
                                         0,0,0,0, 0, 0, 0, 0, 0,
                                                  0, 0, 0, 0, 0/! PALLETZ
                                           0,0,0, 0, 0, 0, 0, 0,
                                                  0, 0, 0, 0, 0/! BEEF_KE
                                             0,0, 0, 0, 0, 0, 0,
                                                  0, 0, 0, 0, 0/! MT_DOCK
                               0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0/! GRAVY_K
                                  0, 0, 0, 0, 0, 0, 0, 0, 0, 0/! GF_DOCK
                                     0, 0, 0, 0, 0, 0, 0, 0, 0/! PEA_DOC
                                        0, 0, 0, 0, 0, 0, 0, 0/! STR_DOCK
                                           0, 0, 0, 0, 0, 0, 0/! PO_DOCK
                                              0, 0, 0, 0, 0, 0/! CA_DOCK
                                                 0, 0, 0, 0, 0/! MATL4_D
                                                    0, 0, 0, 0/!
                                                       0, 0, 0/!
                                                          0, 0/!
                                                             0:! MATL5_D
    2,   19-20,       ! from                       KETTLE ASSISTANT TRANSPORT
           20        :! BEEF_KET
    3,   12-14,       ! from                       RETORT LOADER TRANSPORT
     30,40/           ! BASK_LDR
        30;           ! RETORTS
;
TABLES       :1,1,1,2,3,4,5,6,0,0:        !Product 1    Beef Stew
              2,1,1,2,8,4,0,0,0,0:        !Product 2    Beef Tips
              3,1,1,2,0,4,0,9,0,0:        !Product 3    Beef with Peppers
              4,1,1,10,0,11,12,6,3,13:    !Product 4    Vegetables
              5,1,1,39,.65,17.3,.29,8.8,.15,0,0,0,0,12.6,.2,25.3,4:!PROD 1 FILL
              6,1,1,2.62,1,0,0,0,0,0,0,0,0,5,1,5,1:  !PRODUCT 2 FILLING AMOUNT
              7,1,1,2.62,1,0,0,5,1,0,0,0,0,5,1,5,1:  !PRODUCT 3 FILLING AMOUNT
              8,1,1,5,1,5,1,5,1,5,1,5,1,5,1,5,1;   PRODUCT 4 FILLING AMOUNT
;
;*********************************************************************
;      The conveyor speeds are outlined here
;*********************************************************************
conveyers:                          !in/sec    in/section
    1,TRAYWASH,    1,                24.0000,   12:
    2,GRAVYLIN,    2,                48.0000,    6:
    3,SEAMER  ,    3,                24.0000,   1:
    4,WASH_DRY,    4,                5.4,      1:
    5,filcon,      5,                5.4,      1:
    6,con2,        6,                5.4,      1:
    7,CON3,        7,                5.4,      1:
    8,CON4,        8,                5.4,      1:
    9,CON5,        9,                5.4,      1:
   10,CON6,       10,                5.4,      1:
   11,EXICON,     11,                5.4,      1:
   12,CON7,       12,                5.4,      1:
   13,CON8,       13,                5.4,      1;
;
segments:                    !inches
        1,    4, 5 -   120.0:
        2,   21,22 -   300.0:
        3,   10,11 -   180.0:
        4,   14,15 -   180.0:
        5,    4,5  -   64.0:
        6,    5,6  -   64.0:
        7,    6,7  -   64.0:
```

```
        8,     6,8  -   112.0:
        9,     7,8  -   64.0:
       10,    22,9  -   64.0:
       11,     9,3  -   96.0:
       12,     9,12 -   64.0:
       13,     8,22 -   64.0;
;
;
REPLICATE,    1,            .0000,      ,YES ,YES,3600.0;
;
DSTATS:        1,NR(1),DOCK:
               2,NR(2),STOCK:
               3,NR(3),SCRAP:
               4,NR(4),TR_PLACE:
                5,NR(5),TR_FLIP:
               6,NR(6),POTATOES:
               7,NR(7),MEAT_FIL:
               8,NR(8),CARROTS:
               9,NR(9),GRAVYFIL:
              10,NR(10),CHKWEIGH:
              11,NR(11),FINAL_CW:
              12,NR(12),BASK_LDR:
              13,NR(13),RETORTS:
              14,NR(14),BASK_UNL:
              15,NR(15),TOWEL_WP:
              16,NR(16),VIDEOJET:
              17,NR(17),CASING :
              18,NR(18),PALLETZG:
              19,NR(19),BEEF_KET:
              20,NR(20),MT_DOCK:
              21,NR(21),GRAVY_K:
              22,NR(22),MATL4:
              23,NR(23),MATL5:
              24,NR(24),GRAVYFL2:
              25,NR(25),RETORT1:
              26,NR(26),RETORT2:
              27,NR(27),RETORT3:
              28,X(2),TOP_L:
              29,X(3),BOTTOM_L:
              30,X(4),RETORTS:
              31,NQ(4),TR_PLACE:
              32,NQ(5),TR_FLIP:
              33,NQ(66),GF1:
              34,NQ(7),MAT1:
              35,NQ(67),MAT4:
              36,NQ(68),MAT5:
              37,NQ(6),MAT2:
              38,NQ(8),MAT3:
              39,NQ(9),GF2:
              40,NQ(10),CHK_W:
              41,NQ(11),F_CW:
              42,NQ(12),BL:
              43,x(33),INVEN,22:
              44,x(46),prod,23:
              45,x(41),bot,24:
              46,X(27),SHIPDA,25:
              47,x(43),total,26:
              48,X(19),MATU,27:
```

```
        49,X(39),shipq,28;
;
;
END;
```

```
BEGIN,1,1,YES,prouc,NOIN;
          CREATE,1,10:86450;    Initiate Bottom line
          SIGNAL:3;             with cartoons
          ASSIGN:A(1)=X(50);
          QUEUE,38;
          WAIT:1;
          ROUTE:0,2;
DISP      DELAY:0:DISPOSE;
;
;
;*********************************************************************
;*********************************************************************
;
;                          SCHEDULING MODULE
;
;*********************************************************************
;*********************************************************************
;
;*********************************************************************
;       The input to this routine is the various orders for which the
;       the simulation is to be done. These orders are given in the
;       arrivals block of the experimental file. The parameters would
;       be the Start time of the order, The size of the order, The
;       due date and the kill date beyond which the order can be
;       assumed to be void. This block also starts/stops the machines.
;*********************************************************************
;
;
;
ORDER     QUEUE,48:DETACH;
          QUEUE,1:DETACH;
          CREATE,1:86400;          start daily production
          ASSIGN:X(20)=0;
          ASSIGN:X(17)=0;
          ASSIGN:X(19)=0;
          ASSIGN:X(47)=X(47)+1;
          BRANCH,1:
            IF,NQ(48).NE.0,RE48:
            ELSE,RE1;
RE48      REMOVE:1,48,GETMAT:DISPOSE; Decide product and get material
RE1       BRANCH,1:                   ;reqired.
            IF,NQ(1).NE.0,REM:
            ELSE,END;
REM       REMOVE:1,1,CHKORD;
          DELAY:0.001:DISPOSE;
END       ASSIGN:X(20)=0;
          COUNT:9:DISPOSE;
CHKORD    BRANCH,1:
            IF,TNOW/86400.GE.A(4),REMDIS:
            ELSE,GETMAT;
REMDIS    REMOVE:1,1,CHKORD:DISPOSE;
GETMAT    ASSIGN:X(1)=A(2);
          ASSIGN:X(50)=A(1);
          ASSIGN:X(48)=A(3);
          ASSIGN:X(46)=A(4);
          BRANCH,1:
```

```
          IF,X(1).EQ.-1,FINUP:
          ELSE,BRTYPE;                decide when to stop
FINUP     SIGNAL:3:DISPOSE;
BRTYPE    BRANCH,1:                   ;chosse material based on product type
          IF,A(1).EQ.1,TYPE1:
          IF,A(1).EQ.2,TYPE2:
          IF,A(1).EQ.3,TYPE3:
          IF,A(1).EQ.4,TYPE4;
BEEF      ASSIGN:X(2)=X(2)+20;
          ROUTE:0,1;
POTATOES  ASSIGN:X(3)=20+X(3);
          ROUTE:0,1;
CARROTS   ASSIGN:X(4)=X(4)+20;
          ROUTE:0,1;
STARCH    ASSIGN:X(5)=X(5)+20;
          ROUTE:0,1;
PEAS      ASSIGN:X(6)=X(6)+20;
          ROUTE:0,1;
MUSHROOM  ASSIGN:X(7)=X(7)+20;
          ROUTE:0,1;
BEANS     ASSIGN:X(8)=X(8)+20;
          ROUTE:0,1;
BRINE     ASSIGN:X(9)=X(9)+20;
          ROUTE:0,1;
PEPPER    ASSIGN:X(10)=X(10)+20;
          ROUTE:0,1;
TYPE1     ASSIGN:X(16)=5;
          BRANCH,5:
            ALWAYS,BEEF:
            ALWAYS,POTATOES:
            ALWAYS,CARROTS:
            ALWAYS,PEAS:
            ALWAYS,STARCH;
TYPE2     ASSIGN:X(16)=3;
          BRANCH,3:
            ALWAYS,BEEF:
            ALWAYS,MUSHROOM:
            ALWAYS,STARCH;
TYPE3     ASSIGN:X(16)=3;
          BRANCH,3:
            ALWAYS,BEEF:
            ALWAYS,PEPPER:
            ALWAYS,STARCH;
TYPE4     ASSIGN:X(16)=4;
          BRANCH,4:
            ALWAYS,BRINE:
            ALWAYS,BEANS:
            ALWAYS,PEAS:
            ALWAYS,CARROTS;
;
;
;***************************************************************************
;                        MACHINE CONTROL
;       This block turns the machines off & on depending on the
;       signals from other blocks and  the time of day.
;       It takes into account the diffrence in time between
;       the start of the cooking operations and filling.
;***************************************************************************
```

```
;
        CREATE,1,100:86400;
        QUEUE,88;
        WAIT:3;
        BRANCH,2:
          ALWAYS,CON:
          ALWAYS,INVEN;
CON     ASSIGN:X(33)=0;
        ALTER:FORMER,-1;
        ALTER:SEALER,-1;
        ALTER:CUTTING1,-1;
        ALTER:CUTTING2,-1;
        ALTER:CUTTING3,-1;
        ALTER:SPLITTER,-1;
        ALTER:INSPECT1,-1;
        ALTER:INSPECT2,-1;
        ALTER:INSPECT3,-1;
        ALTER:INSPECT4,-1;
        ALTER:INSPECT5,-1;
        ALTER:INSPECT6,-1;
        ALTER:INSPECT7,-1;
        ALTER:INSPECT8,-1;
        ALTER:LOADMEN(1),-1;
        ALTER:LOADMEN(2),-1;
        ALTER:BLOAD1,-1;
        ALTER:BLOAD2,-1;
        ALTER:INS1,-1;
        ALTER:INS2,-1;
        ALTER:INS3,-1;
        ALTER:INS4,-1;
        ALTER:INS5,-1;
        ALTER:INS6,-1;
        ALTER:INS7,-1;
        ALTER:INS8,-1;
        ALTER:INS9,-1;
        ALTER:INS10,-1;
        ALTER:INS11,-1;
        ALTER:INS12,-1;
        ALTER:INS13,-1;
        ALTER:INS14,-1;
        ALTER:INS15,-1;
        ALTER:INS16,-1;
        ALTER:PACKER1,-1;
        ALTER:PACKER2,-1;
        ALTER:PACKER3,-1;
        ALTER:PACKER4,-1;
        BRANCH,3:
          ALWAYS,Q50:
          ALWAYS,Q96:
          ALWAYS,Q95;
Q50     QUEUE,50;
        SEIZE,2:RETORT(1);
        RELEASE:RETORT(1);
        ALTER:RETORT(1),-1:NEXT(Q94);
Q95     QUEUE,95;
        SEIZE,2:RETORT(2);
        RELEASE:RETORT(2);
        ALTER:RETORT(2),-1:NEXT(Q94);
```

```
Q96       QUEUE,96;
          SEIZE,2:RETORT(3);
          RELEASE:RETORT(3);
          ALTER:RETORT(3),-1:NEXT(Q94);
Q94       QUEUE,94;
          GROUP:3;
          ASSIGN:X(35)=0;
          QUEUE,47;
          WAIT:1;
          ASSIGN:X(33)=1;
          ASSIGN:X(34)=1;
          ALTER:FORMER,1;
          ALTER:SEALER,1;
          ALTER:CUTTING1,1;
          ALTER:CUTTING2,1;
          ALTER:CUTTING3,1;
          ALTER:SPLITTER,1;
          ALTER:INSPECT1,1;
          ALTER:INSPECT2,1;
          ALTER:INSPECT3,1;
          ALTER:INSPECT4,1;
          ALTER:INSPECT5,1;
          ALTER:INSPECT6,1;
          ALTER:INSPECT7,1;
          ALTER:INSPECT8,1;
          ALTER:LOADMEN(1),1;
          ALTER:LOADMEN(2),1;
          ALTER:BLOAD1,1;
          ALTER:BLOAD2,1;
          ALTER:INS1,1;
          ALTER:INS2,1;
          ALTER:INS3,1;
          ALTER:INS4,1;
          ALTER:INS5,1;
          ALTER:INS6,1;
          ALTER:INS7,1;
          ALTER:INS8,1;
          ALTER:INS9,1;
          ALTER:INS10,1;
          ALTER:INS11,1;
          ALTER:INS12,1;
          ALTER:INS13,1;
          ALTER:INS14,1;
          ALTER:INS15,1;
          ALTER:INS16,1;
          ALTER:PACKER1,1;
          ALTER:PACKER2,1;
          ALTER:PACKER3,1;
          ALTER:PACKER4,1;
          ASSIGN:X(35)=1;
          ALTER:RETORT(1),1;
          ALTER:RETORT(2),1;
          ALTER:RETORT(3),1;
          SPLIT;
          DELAY:0:DISPOSE;
;
;**********************************************************************
;**********************************************************************
```

```
;
;                        END SCHEDULING MODULE
;
;*********************************************************************
;*********************************************************************
;
;
;
;
;*********************************************************************
;*********************************************************************
;
;                        INVENTORY MODULE
;
;*********************************************************************
;*********************************************************************
;
;       This block keeps track of the on-hand inventory. The only input
;       is the probability that a given lot might be rejected by the
;       USDA.
;       It gives as output the on-hand inventory at any time
;**********************************************************************;
;
INVEN    ASSIGN:A(1)=D(1);
         ASSIGN:D(2)=D(2)+D(1);
         ASSIGN:D(1)=0;
         ASSIGN:A(2)=X(47);
         BRANCH,1:
           WITH,0.1,USDREJ:
           ELASE,NOREJ
USREJ    ASSIGN:A(2)=A(2)+25;
BR11     BRANCH,1:
           IF,NQ(89) .EQ. 0, SIG9:
           ELSE, REM89;
REM89    REMOVE:1,89,CINVEN;
         QUEUE,87;
         WAIT:9;
         ASSIGN:D(3)=D(2);
Q89      QUEUE,89:DETACH;
CINVEN   BRANCH,1:
           IF, A(2)-X(47) .GE. 10 ,DINV:
           ELSE,BR11;
DINV     ASSIGN:D(2)=D(2)-A(1):DISPOSE;
SIG9     SIGNAL:9:NEXT(Q89);
;
;
;
;*********************************************************************
;*********************************************************************
;
;                        END INVENTORY MODULE
;
;*********************************************************************
;*********************************************************************
;
;
;*********************************************************************
;*********************************************************************
```

```
;
;                         COOKING MODULE
;
;
;**********************************************************************
;**********************************************************************
;
;
;**********************************************************************
;       This is the first operation to start on any given day.
;       It can be divided into two activites Cooking/Cooling
;**********************************************************************
;
         STATION,1;
         QUEUE,2;
         COMBINE:X(16),LAST;
         DUPLICATE:8;
         BRANCH,3:
           ALWAYS,Q37:
           ALWAYS,Q84:
           ALWAYS,Q82;
Q37      QUEUE,37;
         SEIZE:KETTLE;
         DELAY:CO(1);
         RELEASE:KETTLE;
         QUEUE,3;
         SEIZE:COOLER;
         DELAY:CO(2);
         RELEASE:COOLER;
         SIGNAL:1:DISPOSE;
Q84      QUEUE,84;
         SEIZE:KETTLE2;
         DELAY:CO(1);
         RELEASE:KETTLE2;
         QUEUE,83;
         SEIZE:COOLER2;
         DELAY:CO(2);
         RELEASE:COOLER2:DISPOSE;
Q82      QUEUE,82;
         SEIZE:KETTLE3;
         DELAY:CO(1);
         RELEASE:KETTLE3;
         QUEUE,81;
         SEIZE:COOLER3;
         DELAY:CO(2);
         RELEASE:COOLER3:DISPOSE;
         STATION,2;
;
;
;
;**********************************************************************
;**********************************************************************
;
;                         END COOKING MODULE
;
;**********************************************************************
;**********************************************************************
;
```

```
;
;*********************************************************************
;
;                         PCKAGING MODULE
;       The Pouches  are formed, filled, sealed and cut in this
;       module
;
;*********************************************************************
;
;
;
;*********************************************************************
;       The first station here forms the pouches. They are formed in
;       batches of six and move through the packaging module in a group.
;       The group is split only after it reaches the cutting stations
;*********************************************************************
;
;
;
DUP      DUPLICATE:9;
         ASSIGN:X(19)=X(19)+6;
         ASSIGN:A(2)=X(19);
         ASSIGN:A(5)=1;
         DELAY:0.001;
         BRANCH,1:
           IF,TNOW-X(47)*86400.GE.36000.AND.
           X(20).EQ.0,SP:
           ELSE,Q4;
SP       ASSIGN:X(20)=X(19);
         BRANCH,2:
           ALWAYS,Q4:
           IF,X(19).LE.X(1),RET:
           IF,NQ(1).EQ.0.AND.X(1).NE.-1,
           RET1;
RET      ASSIGN:A(2)=X(1)-X(19);
         ASSIGN:A(3)=X(48);
         ASSIGN:A(4)=X(46):NEXT(ORDER);
RET1     ASSIGN:A(2)=-1;
         ASSIGN:A(3)=X(48);
         ASSIGN:A(4)=X(46):NEXT(ORDER);
Q4       COUNT:11;
         QUEUE,49;
         ROUTE:0,3;
         STATION,3;
Q7       QUEUE,4;
         SEIZE:FORMER;
         DELAY:CO(3):MARK(4);
         BRANCH:
           ALWAYS,CONT:
           IF,NQ(4).LE.5.AND.X(20).EQ.0,DUP;
CONT     RELEASE:FORMER;
         ASSIGN:M=4;
         ROUTE:0,4;
;
;
;*********************************************************************
;       Once the pouches are formed they are indexed through the
;       system. The distance between the former and the sealing
```

```
;       station is 12 indexes. This is achieved by the next block.
;       Any number of filling station that can be accomodated
;       in this distance have no bearing on the results since
;       the pouches will be indexed any ways.
;**********************************************************************
;
;
        STATION,4-16;
        DELAY:CO(3),M;
        ASSIGN:M=M+1;
        ROUTE:0,M;
;
;
;**********************************************************************
;       The sealing operationis modeled in the next station.
;       This operation also, is synchronized with rest of the line.
;**********************************************************************
;
;
        STATION,17;
        QUEUE,5;
        SEIZE:SEALER;
        DELAY:CO(3);
        RELEASE:SEALER;
        ROUTE:0,18;
;
;
;**********************************************************************
;       The next three stations model the cutting/spliting operations.
;       The pouches are in two rows and continues. These operations cut
;       them into  individual pouches.
;**********************************************************************
;
;
        STATION,18;
        QUEUE,6;
        SEIZE:CUTTING1;
        ASSIGN:A(5)=10;
        DELAY:CO(3);
        RELEASE:CUTTING1;
        ROUTE:0,19;
        STATION,19;
        QUEUE,7;
        SEIZE:CUTTING2;
        DELAY:CO(3);
        RELEASE:CUTTING2;
        ROUTE:0,20;
        STATION,20;
        QUEUE,8;
        SEIZE:CUTTING3;
        DELAY:CO(3);
        RELEASE:CUTTING3;
        ROUTE:0,21;
        STATION,21;
        QUEUE,9;
        SEIZE:SPLITTER;
        DELAY:CO(3);
        RELEASE:SPLITTER;
```

```
        DUPLICATE:6;
        ASSIGN:A(5)=2;
        ROUTE:0,22;
;*********************************************************************
;*********************************************************************
;
;                        END PACKAGING MODULE
;
;*********************************************************************
;*********************************************************************
;
;
;
;
;*********************************************************************
;*********************************************************************
;
;                        INSPECTION MODULE
;
;*********************************************************************
;*********************************************************************
;
;*********************************************************************
;       In this module the pouches are inspected 100% for bad seals.
;       Only pouches which pass this inspection are retorted.
;*********************************************************************
;
        STATION,22;
        QUEUE,10;
        BRANCH,1:
          WITH,.125,INS1:
          WITH,.125,INS2:
          WITH,.125,INS3:
          WITH,.125,INS4:
          WITH,.125,INS5:
          WITH,.125,INS6:
          WITH,.125,INS7:
          WITH,.125,INS8;
INS1    QUEUE,11;
        SEIZE:INSPECT1;
        DELAY:RN(4,1);
        RELEASE:INSPECT1:NEXT(QUE);
INS2    QUEUE,12;
        SEIZE:INSPECT2;
        DELAY:RN(4,2);
        RELEASE:INSPECT2:NEXT(QUE);
INS3    QUEUE,13;
        SEIZE:INSPECT3;
        DELAY:RN(4,3);
        RELEASE:INSPECT3:NEXT(QUE);
INS4    QUEUE,14;
        SEIZE:INSPECT4;
        DELAY:RN(4,4);
        RELEASE:INSPECT4:NEXT(QUE);
INS5    QUEUE,41;
        SEIZE:INSPECT5;
        DELAY:RN(4,1);
        RELEASE:INSPECT5:NEXT(QUE);
```

```
INS6     QUEUE,42;
         SEIZE:INSPECT6;
         DELAY:RN(4,2);
         RELEASE:INSPECT6:NEXT(QUE);
INS7     QUEUE,43;
         SEIZE:INSPECT7;
         DELAY:RN(4,3);
         RELEASE:INSPECT7:NEXT(QUE);
INS8     QUEUE,44;
         SEIZE:INSPECT8;
         DELAY:RN(4,4);
         RELEASE:INSPECT8:NEXT(QUE);
QUE      TALLY:A(1),INT(4);
         COUNT:A(1);
         BRANCH,1:
           WITH,0.5,Q93:
           WITH,0.5,Q92;
Q93      QUEUE,93;
         SEIZE:BLOAD1;
         DELAY:RN(8,1);
         RELEASE:BLOAD1:NEXT(EL);
Q92      QUEUE,92;
         SEIZE:BLOAD2;
         DELAY:RN(8,1);
         RELEASE:BLOAD2:NEXT(EL);
EL       ASSIGN:X(30)=X(30)+1;
         BRANCH,1:
           IF,X(30).EQ.X(20),SETX22:
           ELSE,Q15;
SETX22   ASSIGN:X(22)=NQ(15)+1;
         ASSIGN:A(2)=X(20);
         ASSIGN:X(30)=0;
Q15      QUEUE,15;
         COMBINE:X(22),LAST;
         BRANCH,1:
           IF,A(2).EQ.X(20),SETX23:
           ELSE,Q16;
SETX23   ASSIGN:X(23)=NQ(16)+1;
Q16      ASSIGN:A(5)=4;
         QUEUE,16;
         COMBINE:X(23),LAST;
         BRANCH,1:
           IF,A(2).EQ.X(20),SETX24:
           ELSE,Q17;
SETX24   ASSIGN:X(24)=NQ(17)+1;
Q17      ASSIGN:A(5)=5;
         QUEUE,17;
         COMBINE:X(24),LAST;
         ASSIGN:
           A(3)=X(22)+(X(23)-1)*8+(X(24)-1)*
           64;
         ASSIGN:X(22)=8;
         ASSIGN:X(23)=8;
         ASSIGN:X(24)=16;
         ASSIGN:A(5)=6;
;
;
;*********************************************************************
```

```
;*********************************************************************
;
;                        END INSPECTION MODULE
;
;*********************************************************************
;*********************************************************************
;
;
;
;*********************************************************************
;*********************************************************************
;
;                        RETORTING MODULE
;
;*********************************************************************
;*********************************************************************
;
;
;
;*********************************************************************
;       The pouches are grouped in batches of 1024. Two of these
;       batches can be loaded in a retort at a time.
;*********************************************************************

         STATION,23;
         QUEUE,59;
         REQUEST,1,1:CART;
         TRANSPORT:CART,24;
         STATION,24;
         FREE:CART;
         BRANCH,1:
           IF,A(2).EQ.X(20),SETX25:
           ELSE,SETA33;
SETX25   ASSIGN:X(25)=NQ(18)+1;
         SIGNAL:3;
SETA33   ASSIGN:A(3)=A(3)+NQ(18)*1024;
         QUEUE,18;
         COMBINE:X(25),LAST;
         ASSIGN:X(25)=2;
         COUNT:13;
         QUEUE,19;
         SELECT,RAN:
          RETORT1:
          RETORT2:
          RETORT3;
RETORT1  SEIZE, 1:   RETORT(1);
         ASSIGN:A(4)=1:NEXT(RETREL);
RETORT2  SEIZE, 1:   RETORT(2);
         ASSIGN:A(4)=2:NEXT(RETREL);
RETORT3  SEIZE, 1:   RETORT(3);
         ASSIGN:A(4)=3:NEXT(RETREL);
RETREL   ASSIGN:     A(5)= 9;          change to 4 basket symbol
         DELAY:      CO(5);
         RELEASE:    RETORT(A(4)) ;
         QUEUE,40;
         SCAN:TNOW-X(47)*86400.LT.36000;
         QUEUE,60;
         REQUEST,1:CART;
```

```
        TRANSPORT:CART,25;
;
;
;
;
;**********************************************************************
;**********************************************************************
;
;                          END RETORTING MODULE
;
;**********************************************************************
;**********************************************************************
;
;
;
;
;**********************************************************************
;**********************************************************************
;
;                          CASING MODULE
;
;**********************************************************************
;**********************************************************************
;
;
;
;**********************************************************************
;       The pallets are removed from the retort and sent over to this
;       module. The pouches are removed from the pallets and cased.
;       The various operations involved are washed, inspected,
;       labeling and cartooning;
;**********************************************************************
;
;
;**********************************************************************
;       This station unloads the pouches from the pallet and places
;       them on the conveyor where it is washed.
;**********************************************************************
;
;
         STATION,25;
         FREE:CART;
         ASSIGN:X(18)=X(18)+A(3);
DUP1     DUPLICATE:9;
         ASSIGN:A(5)=7;
         ASSIGN:X(18)=X(18)-1;
         BRANCH,1:
           IF,X(18).LT.0,DIS:
           ELSE,QUEUP;
DIS      ASSIGN:X'18)=X(18)+1:DISPOSE;
QUEUP    QUEUE,20;
         SELECT,CYC:
           L1:
           L2;
L1       SEIZE,2:LOADMEN(1);
         ASSIGN:A(5)=2;
         ASSIGN:A(4)=1:NEXT(LREL);
L2       SEIZE,2:LOADMEN(2);
```

```
         ASSIGN:A(5)=2;
         ASSIGN:A(4)=2:NEXT(LREL);
LREL     COUNT:14;
         DELAY:RN(8,1);
         BRANCH:
           ALWAYS,CONTI1:
           IF,NQ(20).LE.2.AND.X(18).GT.0,
           DUP1;
CONTI1   RELEASE:LOADMEN(A(4));
         QUEUE,21;
         ACCESS:CONV3;
         CONVEY:CONV3,26;
;
;
;*************************************************************************
;        This is the final inspection station. The pouches are
;        inspected for any possible defect. This is also a 100%
;        inspection.
;*************************************************************************
         STATION,26;
         EXIT:CONV3:MARK(4);
         BRANCH,8:
           WITH,1/16,IN1:
           WITH,1/16,IN2:
           WITH,1/16,IN3:
           WITH,1/16,IN4:
           WITH,1/16,IN5:
           WITH,1/16,IN6:
           WITH,1/16,IN7:
           WITH,1/16,IN8:
           WITH,1/16,IN9:
           WITH,1/16,IN10:
           WITH,1/16,IN11:
           WITH,1/16,IN12:
           WITH,1/16,IN13:
           WITH,1/16,IN14:
           WITH,1/16,IN15:
           WITH,1/16,IN16;
IN1      QUEUE,22;
         SEIZE,2:INS1;
         DELAY:RN(7,7);
         RELEASE:INS1:NEXT(QUE3);
IN2      QUEUE,23;
         SEIZE,2:INS2;
         DELAY:RN(7,8);
         RELEASE:INS2:NEXT(QUE3);
IN3      QUEUE,24;
         SEIZE,2:INS3;
         DELAY:RN(7,7);
         RELEASE:INS3:NEXT(QUE3);
IN4      QUEUE,25;
         SEIZE,2:INS4;
         DELAY:RN(7,9);
         RELEASE:INS4:NEXT(QUE3);
IN5      QUEUE,26;
         SEIZE,2:INS5;
         DELAY:RN(7,1);
         RELEASE:INS5:NEXT(QUE3);
```

```
IN6       QUEUE,27;
          SEIZE,2:INS6;
          DELAY:RN(7,2);
          RELEASE:INS6:NEXT(QUE3);
IN7       QUEUE,28;
          SEIZE,2:INS7;
          DELAY:RN(7,3);
          RELEASE:INS7:NEXT(QUE3);
IN8       QUEUE,29;
          SEIZE,2:INS8;
          DELAY:RN(7,7);
          RELEASE:INS8:NEXT(QUE3);
IN9       QUEUE,31;
          SEIZE,2:INS9;
          DELAY:RN(7,7);
          RELEASE:INS9:NEXT(QUE3);
IN10      QUEUE,32;
          SEIZE,2:INS10;
          DELAY:RN(7,8);
          RELEASE:INS10:NEXT(QUE3);
IN11      QUEUE,33;
          SEIZE,2:INS11;
          DELAY:RN(7,7);
          RELEASE:INS11:NEXT(QUE3);
IN12      QUEUE,35;
          SEIZE,2:INS12;
          DELAY:RN(7,9);
          RELEASE:INS12:NEXT(QUE3);
IN13      QUEUE,90;
          SEIZE,2:INS13;
          DELAY:RN(7,1);
          RELEASE:INS13:NEXT(QUE3);
IN14      QUEUE,39;
          SEIZE,2:INS14;
          DELAY:RN(7,2);
          RELEASE:INS14:NEXT(QUE3);
IN15      QUEUE,45;
          SEIZE,2:INS15;
          DELAY:RN(7,3);
          RELEASE:INS15:NEXT(QUE3);
IN16      QUEUE,46;
          SEIZE,2:INS16;
          DELAY:RN(7,7);
          RELEASE:INS16:NEXT(QUE3);
QUE3      QUEUE,30;
          ACCESS:CONV4;
          ASSIGN:X(49)=A(1);
          CONVEY:CONV4,27;
;
;
;*********************************************************************
;       The pouches are individually packed and then cartooned.
;       These operations are modeled in the next two stations.
;*********************************************************************
;
          STATION,27;
          EXIT:CONV4;
          BRANCH,1:
```

```
            WITH,.25,PACKER1:
            WITH,.25,PACKER2:
            WITH,.25,PACKER3:
            WITH,.25,PACKER4;
PACKER1   QUEUE,34;
          SEIZE:PACKER1;
          ASSIGN:A(5)=10;
          DELAY:RN(8,1);
          RELEASE:PACKER1:NEXT(QUE56);
PACKER2   QUEUE,72;
          SEIZE:PACKER2;
          ASSIGN:A(5)=10;
          DELAY:RN(8,2);
          RELEASE:PACKER2:NEXT(QUE56);
PACKER3   QUEUE,85;
          SEIZE:PACKER3;
          ASSIGN:A(5)=10;
          DELAY:RN(8,1);
          RELEASE:PACKER3:NEXT(QUE56);
PACKER4   QUEUE,86;
          SEIZE:PACKER4;
          ASSIGN:A(5)=10;
          DELAY:RN(8,2);
          RELEASE:PACKER4;
QUE56     QUEUE,56;
          ACCESS:CONV6;
          ASSIGN:A(5)=15;
          CONVEY:CONV6,28;
;
;
          STATION,28;
          EXIT:CONV6;
          ASSIGN:D(1)=D(1)+1;
          ASSIGN:A(1)=A(1)+4;
          TALLY:A(1),INT(4);
          COUNT:A(1);
          ASSIGN:A(1)=A(1)-4;
          BRANCH,1:
            IF,A(1).EQ.X(40).OR.NQ(36).EQ.0,
            CON12:
            ELSE,SETX26;
SETX26    ASSIGN:X(26)=NQ(36);
CON12     ASSIGN:X(40)=A(1);
          QUEUE,36;
          COMBINE:X(26),LAST;
          ASSIGN:X(26)=20:DISPOSE;
END;

;*********************************************************************
;*********************************************************************
;
;                        END CASING MODULE
;
;*********************************************************************
;*********************************************************************
;
;
;
```

```
;
;**********************************************************************
;**********************************************************************
;
;                         END OF THE MODEL FILE
;
;**********************************************************************
;**********************************************************************;
;
;
;
;
;
;**********************************************************************
;**********************************************************************

                           EXPERIMENTAL FILE

;**********************************************************************
;**********************************************************************
;
;
;**********************************************************************
;**********************************************************************
;
;
;**********************************************************************
;       This file contains  the data required to run the model.
;       Some of this data is user input but a good part of it
;       is the data required by siman. It also has the definition
;       of the various stations, resources etc used in the model.
;**********************************************************************


BEGIN         ,10,10,NO,NO;
;
;
PROJECT       ,POUCH,JAFARI.M,1/25/1990;
;
DISCRETE      ,650,5,99,30,5;
;
;**********************************************************************
;       The arrivals block stores the various orders that have to
;       be processed.
;       1(order #),QUEUE(80)(queue #),0(Time when orders is to be created)
;       1(batch size of the orders),2( product type),21856(order size)
;       ,1(due date, in days),9999(kill date , in days):
;**********************************************************************************
**************************************************
ARRIVALS     :1,QUEUE(88),0.00,1:
              2,QUEUE(1),0.00,1,1,50000,0.00,100000;
;
;**********************************************************************
;       The block intializes certain varibles like pallet size
;**********************************************************************
INITIALIZE  ,X(17)=0,
             X(19)=0,
             X(20)=0,
```

```
             X(47)=-1,
             X(21)=8,                  !
             X(22)=8,                  !palet size 8*8*16
             X(23)=8,                  !
             X(24)=16,
             X(25)=2,
             X(26)=20,
             X(18)=0;
;
PARAMETERS  :1,1800:                      !CO>Cook time
             2,1800:                      !CO>Cooling time
             3,4:                         !CO>casing & paletizing

             4,4,.25:                     !RN>
             5,3600:                      !CO>Retort Cycle time
             6,20:                        !CO>
             7,7,0:                       !RN>
             8,1,0:                       !RN>
             9,10:                        !co> Incubation period
            10,0.1:                       !co> USDA rejection
;
RANKINGS    :1-1,LVF(3):
             2-40,FIFO;
;***************************************************************************
;       This gives a list of resources used in the model.
;***************************************************************************
RESOURCES   :1-1,KETTLE,1:
             2-2,COOLER,1:
             3-3,FORMER,1:
             4-4,SEALER,1:
             5-5,CUTTING1,1:
             6-6,CUTTING2,1:
             7-7,CUTTING3,1:
             8-8,SPLITTER,1:
             9-9,INSPECT5,1:
             10-10,INSPECT6,1:
             11-11,INSPECT7,1:
             12-12,INSPECT8,1:
             13-13,INS9,1:
             14-14,INS10,1:
             15-15,INSPECT1:
             16-16,INSPECT2:
             17-17,EMPTY:
             18-18,INSPECT3:
             19-19,INSPECT4:
             20-20,EMPTY:
             21-21,INS11,1:
             22-22,INS1:
             23-23,INS2:
             24-24,INS3:
             25-25,INS4:
             26-26,INS5,1:
             27-27,INS6,1:
             28-28,INS7,1:
             29-29,INS8,1:
             30-30,INS12,1:
             31-31,PACKER1,1:
             32-32,PACKER2,1:
```

```
            33-33,INS13:
            34-34,INS14:
            35-35,INS15:
            36-36,INS16:
            37-39,RETORT,1,1,1:
            40-40,BLOAD1:
            41-41,BLOAD2,1:
            42-43,LOADMEN,1,1:
            44-44,PACKER3,1:
            45-45,PACKER4,1:
            46-46,RETOP,1:
            47-47,KETTLE2,1:
            48-48,KETTLE3,1:
            49-49,COOLER2,1:
            50-50,COOLER3,1;
;
;*********************************************************************
;       The speed of the transporter is given in this block
;*********************************************************************
TRANSPORTERS:1,CART,1,1,1.00,23-A;
;
DISTANCES   :1,23-25,
               10,   10/
                     10;
;*********************************************************************
;       The conveyor speeds are outlined here
;*********************************************************************
CONVEYORS   :1,CONV3,1,24,6,ACTIVE:
             2,CONV4,2,24,6,ACTIVE:
             3,CONV6,3,24,6,ACTIVE;
;
SEGMENTS    :1,25,26-72:
             2,26,27-60:
             3,27,28-80;
;*********************************************************************
;       The  following decide the information that is to be put in
;       the siman output report. Most of this information is to
;       ensure that the model went through its noraml execution.
;*********************************************************************;
DSTAT       :1,NR(3),UTIL OF FEEDER:
             2,NR(4),YTIL OF FILLER:
             3,NR(5),UTIL OF SEALMAN:
             4,NR(6),YTIL OF SEALING:
             5,NR(7),UTIL OF INSPECT1:
             6,NR(8),YTIL OF INSPECT2:
             7,NR(9),UTIL OF INSPECT3:
             8,NR(10),YTIL OF INSPECT4:
             9,X(33),TOPLINE:
             10,X(34),BEEFFEED:
             11,X(35),RETORT;
;
TALLIES     :1,T BEFORE RET P1:
             2,T BEFORE RET P2:
             3,T BEFORE RET P3:
             4,T BEFORE RET P4:
             5,T PACKING P1:
             6,T PACKING P2:
             7,T PACKING P3:
```

```
               8,T PACKING P4;
;
COUNTERS      :1,NO FILLED P1,,YES:
               2,NO FILLED P2,,YES:
               3,NO FILLED P3,,YES:
               4,NO FILLED P4,,YES:
               5,NO PACKED P1,,YES:
               6,NO PACKED P2,,YES:
               7,NO PACKED P3,,YES:
               8,NO PACKED P4,,YES:
               9,END SIMU,1,YES:
               10,BEEFFEED,,YES:
               11,FEEDER,,YES:
               12,SEALMAN,,YES:
               13,RETORT,,YES:
               14,LOADMAN,,YES;
;
OUTPUT        :1,D(3),10,INVENTORY;
;
REPLICATE     ,1,0.0,,Y,Y,0.0;
;
END;
```

COMBAT RATION ADVANCED MANUFACTURING TECHNOLOGY DEMONSTRATION (CRAMTD)

Preliminary Database Design for the CRAMTD Demonstration Plant

Technical Working Paper (TWP) 56

N.R. Adam, T.O. Boucher, T. Chamberlin, J. McPhail, and J.M. Weber
CRAMTD, Food Science Building
Rutgers, The State University of New Jersey

July 1992

Sponsored by:
DEFENSE LOGISTICS AGENCY
Cameron Station
Alexandria, VA 22304-6145

Contractor:
Rutgers, The State University of New Jersey
THE CENTER FOR ADVANCED FOOD TECHNOLOGY*
Cook College
N.J. Agricultural Experiment Station
New Brunswick, New Jersey 08903

DR. JACK L. ROSSEN
Program Director, CRAMTD
DR. JOHN F. COBURN
Associate Director, CRAMTD

TEL: 908-932-7985/8307
FAX: 908-932-8690

*A New Jersey Commission on Science and Technology Center

"Preliminary Database Design for the CRAMTD Demonstration Plant"

Technical Working Paper (TWP) 56

N.R. Adam, T.O. Boucher, T. Chamberlin, J. McPhail, and J. Weber

Abstract

This technical report summarizes the work on the database selected (ORACLE) to implement the Informational Architecture for Packaged Food Manufacture. The report Appendices include: the database tables showing the correspondence between the ORACLE database management system and the IDEF1X Informational Architecture, SQL forms both fully and partially implemented (forms are related to the Functional Architecture), and SQL reports with the identification of supported functions.

1.0 Introduction

This report addresses the requirements of Task Item 3.7.5 of STP #4, requiring a Technical Report on the design of a preliminary database for the CRAMTD Plant. Phase II of STP #4 required studying the procedur ch coalition companies operated their enterprises in the manufacture of shelf stable food products. Based on these studies the research team abstracted the common features of the coalition companies studied, developing a generic set of operating procedures. This generic set is referred to as a "Functional Architecture". A Functional Architecture is a description of the functions performed in operating the enterprise and the relationship among those functions as given by the information flows and material flows linking them. The Functional Architecture was published as Technical Working Paper (TWP)37, "Technical Report: Functional Architecture for Packaged Food Manufacture".

Phase III of STP #4 required identifying the data requirements necessary to support the activities modeled by the Functional Architecture. These data requirements are modeled using an entity-attribute-relationship methodology developed under sponsorship of the U.S. Air Force. This methodology, called IDEF1X (Integrated Computer-Aided Manufacturing Definition 1, Extended), allows the user to create a logical relational database design that is easily understood by business professionals without computer training. This model of the data requirements and their relationships is called an "Informational Architecture". The

Informational Architecture was published as Technical Working Paper (TWP)52, "Technical Report: Informational Architecture for Packaged for Food Manufacture".

Phase IV of STP #4 required implementing the Informational Architecture in a physical database system. This database was to be preliminary in nature thus providing a basis for experimental prototyping and testing of screens and reports that could be implemented in the CRAMTD Phase II demonstration facility. The preliminary database that was developed has established the foundation for final database design and implementation in CRAMTD Phase II. It is the purpose of this report to summarize the work completed on this Task Item in STP #4.

2. Relational Databases and the ORACLE Database Management System

The preliminary database has been implemented on an ORACLE database running on an IBM PS2/70. ORACLE is a database management system (DBMS that is based on the relational model. According to this model, the relational database is made up of a set of interrelated two-dimensional tables where atomic values are stored in any table cell. A DBMS that is based on the relational model should provide a query language that is based on relational algebra or relational calculus. ORACLE uses SQL (Structured Query Language) as its query language. SQL is considered the standard relational query language in the database industry.

In addition to the SQL, ORACLE provides additional tools including the following:

1. SQL*Plus. Enables users who are familiar with SQL to query the database and perform data definition, manipulation, and control operations.

2. SQL*Forms. Enables application developers to develop a user-friendly forms-driven interface to an ORACLE application. Sophisticated forms can be easily developed by including SQL statements in the form of Triggers.

3. SQL*Menu. Enables application developers to integrate various ORACLE forms, reports and other ORACLE functions into a menu-driven application.

4. SQL*ReportWriter. Enables application developers to use a menu-driven tool for generating complex reports using SQL statements.

5. SQL*Calc and SQL*Graph. Provide spreadsheet and graphics interface to an ORACLE database.

6. ProC. Provides the user an interface for applications programs written in the C programming language.

In this report we will show the current status of development work using ORACLE DBMS. For more specific information on the ORACLE tool set, the reader is referred to the references at the end of the text of this report.

3. Tables and Their Relationship to IDEF1X

The basic record keeping element in a relational database is the Table. A table is created for each element of the enterprise about which we want to keep information. The table may be defined

for something tangible; e.g., an employee, or for something abstract; e.g., a production schedule.

The Table of a database corresponds exactly to the concept of an Entity in an IDEF1X model. In fact, database tables are created by implementing IDEF1X entities in software. Exhibit A.1, Appendix A, is a list showing the correspondence between the table names in the ORACLE DBMS and the IDEF1X Entity names reported in reference [1].

Tables have attributes, which are the characteristics of the entity about which information is to be kept. So, for example, a Table of Employees will contain attributes of Employee-id, Employee-Name, Employee Hourly Rate, etc. Each record, or instance of an employee will be described by those attributes. At least one attribute of the table should be unique for each record. So, for example, Employee-id uniquely identifies a particular employee.

In Appendix A, Exhibit A.2, we have summarized all the tables existing in the preliminary database. The reader will note the correspondence between the attributes of the tables and the attributes of their corresponding Entities of the IDEF1X model. There are some additional tables implemented in the database that are system tables used for database management purposes. These have no corresponding entities in IDEF1X.

4. User Screens and SQL Forms

This brief description will assist the reader in understanding

the tables in Appendix A. For further information, the reader is referred to references [4] and [6].

One purpose of database design is to provide a user friendly interface for entering and retrieving data. The user should be able to query the database regarding the status of the enterprise operations; for example, the status of a customer order or the status of the raw ingredient inventory. The user should also be able to incorporate status changes into the database; for example, changes in finished product inventory due to recent shipments or new finished production. SQL Forms provides a "window" to the database tables that allows the user to query or update the tables.

Forms must be designed by the system designers. Several forms have been designed for the preliminary database and are shown in Appendix B. There are two categories of forms in that Appendix: 1) forms that are fully implemented and 2) forms that are partially implemented. A fully implemented form has been designed in considerable detail and usually includes "triggers" that query multiple tables and often run validity checks on a data entry. For example, if a user is entering a new delivery of raw material, the system should check outstanding purchase orders to insure that this raw material has been ordered. The system should also close out the purchase order. Some SQL forms have been implemented at this level of detail.

Partially implemented forms provide a screen for displaying records from tables, but do not have the level of detailed implementation as a fully implemented form.

The forms of Appendix B are displayed in two formats. The first format is shown in Exhibit B.1. The upper left hand corner displays the information that allows the user to relate the form to the Functional Architecture of reference [2]. The Functional Architecture describes the functions that are performed in operating the enterprise. The database is used to support those functions. The entry at the upper left allows the user to relate the screen to the functions described in reference [2].

Below the functional context is the screen itself as it would appear to the user. No data is presented on these example screens.

Finally, below the screen is detail concerning SQL code used in the implementation of the screen. The purpose of the SQL code is shown on the left hand side of the page; the SQL code itself is shown on the right hand side of the page. For further interpretation the reader is referred to reference [3].

The second kind of format is illustrated by Exhibit B.2. Here we do not define a relationship to the Functional Architecture, nor do we define SQL codes for Triggers. These are not user screens. These forms are used by the Database Administrator to enter data directly and are used by the system development team to put test data into the database.

The process of creating user screens using SQL forms will continue through Phase II of CRAMTD as the preliminary database design evolves into a final database implementation.

5. User Reports

Another vehicle for obtaining information from the database is the use of SQL ReportWriter. Whereas SQL Forms allows user interaction in both an input and query mode, SQL ReportWriter only provides fixed format output. When it is desirable to generate hard copy reports, SQL ReportWriter is used.

The SQL reports developed under this task item are shown in Appendix C. As in the case of SQL forms we have formatted the pages to indicate the functions the report is intended to support in the upper left hand corner. This is followed by the report format and, finally, by the SQL routine that queries the tables and formats the data. For further information on SQL ReportWriter, the reader is referred to reference [5].

Summary

In this technical report we have outlined the work performed under STP #4 in designing a preliminary database for CRAMTD. This work is ongoing and represents a starting point for Phase II prototyping and implementation. The task of database development and validity testing will not be complete until all user screens are implemented and the software has been tested by a structured walk through that simulates the actual operation of the enterprise, as outlined in the Functional Architecture. These tasks will be undertaken in CRAMTD Phase II.

References:

1. Adam, N.R., T.O. Boucher, T. Chamberlin and J. Weber, "Technical Report: Informational Architecture for Packaged Food Manufacturing", Technical Working Paper (TWP) 52.

2. Boucher, T.O., M.A. Jafari, S. Kim and J. McPhail, "Technical Report: Functional Architecture for Packaged Food Manufacturing", Technical Working Paper (TWP) 37.

3. "SQL Forms Designers Reference", Version 2.3, Oracle Corporation, 1988.

4. "SQL Language Reference Manual; Version 5.1, Oracle Corporation, 1988.

5. "SQL ReportWriter Reference Manual", Version 1'.0, Oracle Corporation, 1990.

6. "SQL Plus Reference Manual", Version 2.1, Oracle Corporation, 1988.

APPENDIX A

DATABASE TABLES

Exhibit A.1

TABLE NAME	CORRESPONDING IDEF1X BLOCK
ACCOUNT_R	ACCOUNTS RECEIVABLE
BATCH_D	BATCH DETAIL
BATCH_RECORD	BATCH RECORD
CUSTOMER	CUSTOMER
CUSTOMER_ORDER	CUSTOMER ORDER
CUSTOMER_ORDER_D	CUSTOMER DETAIL
DEPARTMENT	DEPARTMENT
EMPLOYEE	EMPLOYEE
FILLING_RECORD	FILLING RECORD
INVOICE_PAY	INVOICE PAYABLE
LABEL	LABEL
LABOR_TICKET	LABOR TICKET
LINE_PROCESS	PROD LINE/PROCESS XREF
MACHINE	MACHINE
MACHINE_AVAIL	MACHINE AVAILABLE
MACHINE_PROCESS	MACHINE/PROCESS XREF
MACHINE_SKILL	MACHIN/SKILL
MATERIAL	MATERIAL
MATERIAL_LIST	MATERIAL LIST
MATERIAL_MOVE	MATERIAL MOVE
MAT_PURPLAN	MATERIAL PURCHASE PLAN
MATERIAL_SPEC	MATERIAL SPEC
PALLET_CARD	PALLET CARD
PALLET_D	PALLET DETAIL
PM_HISTORY	PM HISTORY
PO	PURCHASE ORDER
PO_ITEM	PO ITEM
PRICE_BREAK	PRICE BREAK
PROCESS	PROCESS
PRODSEL	APPROVED PRODUCT LIST
PRODUCT	PRODUCT
PRODUCTION_LINE	PRODUCTION LINE
QUALITY_REPORT	QUALITY REPORT
QUOTE	QUOTE
QUOTE_D	QUOTE DETAIL
RECIPE	RECIPE
RECIPE_LINE	RECIPE/LINE XREF
RECIPE_MASTER	RECIPE MASTER
REQUISITION	REQUISITION
RETORT	RETORT
RFQ	REQUEST FOR QUOTATION
SCHEDULE_D	SCHEDULE DETAIL
SCHEDULE_M	SCHEDULE MASTER
SHIPMENT	SHIPMENT
SHIP_CROSS_REF	SHIPMENT CROSS REF
SKILL	SKILL
SOLICIT	SOLICITATION
TASK	TASK
TEST	TESTS
TEST_SAMPLE	TEST SAMPLE
VENDOR	VENDOR
VENDOR_LOT	VENDOR LOT
VENDOR_QUOTE	VENDOR QUOTE
VENXREF	VENDOR CROSS REF
WO_PARTS	WO PARTS DETAIL
WO_TASK_EMP	WO TASK/EMP DETAIL
WORK_ORDER	WORK ORDER

Exhibit A.2

TABLE_NAME	ATTRIBUTE_NAME
ACCOUNT_R	CUST_INV_NO
	CUST_LINE_NO
	ORDER_NO
	SHIPMENT_NO
BATCH_D	BATCH_QTY
	BATCH_START_HOUR
	BATCH_START_MIN
	FILLING_LINE
	MATERIAL_LOT_NO
	PRODUCTION_DATE
	PRODUCT_ID
BATCH_RECORD	BATCH_SIZE
	BATCH_START_HOUR
	BATCH_START_MIN
	COOK_TEMP
	COOK_TIME_HOUR
	COOK_TIME_MIN
	FILLING_LINE
	KETTLE_ID
	PRODUCTION_DATE
	PRODUCT_ID
	RECIPE_ID
	SUPERVISOR_EMP_ID
CUSTOMER	CUST_ADDR1
	CUST_ADDR2
	CUST_CITY
	CUST_EXT
	CUST_ID
	CUST_NAME
	CUST_PHONE
	CUST_STATE
	CUST_ZIP
CUSTOMER_ORDER	CUST_ID
	CUST_PO
	EMP_ID
	ORDER_DATE
	ORDER_NO

CUSTOMER_ORDER_D	CUST_LINE_NO CUST_LINE_STATUS CUST_QTY CUST_REQUEST_DATE DUMMY EFFECTIVE_DUE_DATE LABEL_ID LAST_PRIORITIZED_DATE ORDER_NO PACKING_QTY PRICE PRIORITY PRODUCTION_LINE_ID PRODUCTION_QTY PRODUCT_ID RECIPE_ID
DEPARTMENT	DEPT_ID DEPT_NAME DEPT_PHONE
EMPLOYEE	EMP_CITY EMP_FNAME EMP_ID EMP_LNAME EMP_PHONE EMP_RATE EMP_STATE EMP_STREET EMP_ZIP SKILL_ID
FILLING_RECORD	FILLING_LINE MATERIAL_LOT_NO PRODUCT_ID PRODUTION_DATE QTY_FILLED
INVOICE_PAY	INVOICE_NO INVOICE_QTY MATERIAL_LOT_NO PO_ITEM PO_NO

LABEL	LABEL_ID LABEL_NAME
LABOR_TICKET	CUST_LINE_NO EMP_ID ORDER_NO SHIFT_NO WORK_DATE WORK_HOURS
LINE_PROCESS	OPERATION_SEQ PROCESS_ID PRODUCTION_LINE_ID
MACHINE	MACHINE_DESC MACHINE_ID MACHINE_LABOR MACHINE_LOCATION
MACHINE_AVAIL	AVAIL_TIME MACHINE_ID PERIOD_LENGTH
MACHINE_PROCESS	MACHINE_ID PROCESS_ID
MACHINE_SKILL	LABOR_QTY MACHINE_ID SKILL_ID
MATERIAL	LAST_BUY LAST_PAID MATERIAL_DESC MATERIAL_ID MATERIAL_UOM REORDER_POINT REORDER_QTY STD_COST
MATERIAL_LIST	AMT_BY_UNIT MATERIAL_ID PRODUCT_ID RECOVERY_PERCENT UNIT_OF_MEASURE

MATERIAL_MOVE	FILLING_LINE MATERIAL_LOT_NO MOVE_QTY MOVE_TO_LOCATION PRODUCTION_DATE PRODUCT_ID TRANSACTION_TYPE
MATERIAL_SPEC	AQL HIGH_LIMIT INSPECTION_PROCEDURE LOW_LIMIT MATERIAL_ID
MAT_PURPLAN	ACCEPT_QTY MATERIAL_ID PRODUCT_ID RECIPE_ID SOLICIT_ID VENDOR_ID
PALLET_CARD	CUST_LINE_NO LABEL_ID LOCATION ORDER_NO ORIG_CUST_LINE_NO ORIG_ORDER_NO PALLET_ID PALLET_STATUS PRODUCT_ID VAR_DATE VAR_STATUS VAR_UNITS
PALLET_D	COOK_NO FILLING_LINE LABEL_DATE PALLET_ID PRODUCTION_DATE QTY_CASES RETORT_ID

PM_HISTORY	FREQ LAST_DATE MACHINE_ID TASK_ID
PO	PO_DATE PO_NUMBER PO_QUANTITY PO_STATUS REQUISITION_NO VENDOR_ID
PO_ITEM	MATERIAL_ID PO_ITEM PO_ITEM_BAL PO_ITEM_DELIVER PO_ITEM_PRICE PO_ITEM_QTY PO_ITEM_STATUS PO_NO
PRICE_BREAK	BREAK_PRICE BREAK_QTY PRODUCT_ID
PROCESS	PROCESS_CLASS PROCESS_DESC PROCESS_ID
PRODSEL	PRODSEL_DUE PRODUCT_ID SELECT_QTY SOLICIT_ID
PRODUCT	CAN_SIZE CAN_SPEC LID_SPEC NET_WEIGHT PRODUCT_ID PRODUCT_NAME QTY_PER_CASE REJECT_RATE REWORK_RATE VALID_TILL

PRODUCTION_LINE	PRODUCTION_LINE_DESC PRODUCTION_LINE_ID
QUALITY_REPORT	MATERIAL_ID MATERIAL_LOT_NO TEST_ID TEST_RESULT
QUOTE	CUST_ID EXPERATION_DATE QUOTE_DATE QUOTE_ID
QUOTE_D	PRODUCT_ID QUOTE_ID QUOTE_LINE QUOTE_PRICE QUOTE_QTY QUOTE_SHIP_DATE
RECIPE	MATERIAL_ID PRODUCT_ID RECIPE_ID
RECIPE_LINE	PRODUCTION_LINE_ID PRODUCT_ID RATE RECIPE_ID
RECIPE_MASTER	AMT_END_UNITS CSIZE GALLONS PROCESS_TEMP PROCESS_TIME PRODUCT_ID RECIPE_ID TARGET_COOK_TEMP TARGET_COOK_TIME TARGET_INIT_TEMP TYPE_COOK

REQUISITION	DEPT MATERIAL_ID REQ_DATE REQ_ID REQ_QTY REQ_STATUS UNIT_OF_MEASURE
RETORT	COOK_NO DISPOSITION END_COMEUP_HOUR END_COMEUP_MIN END_COMEUP_TEMP END_COOK_HOUR END_COOK_MIN END_COOK_TEMP END_FILL_HOUR END_FILL_MIN END_VENT_HOUR END_VENT_MIN END_VENT_TEMP FILLING_LINE INCUBATION_END INCUBATION_START INITIAL_TEMP INSPECTOR_EMP_ID LEAD_EMP_ID NO_CARTS NO_SAMPLES PRODUCTION_DATE PRODUCT_ID RETORT_ID RETORT_QTY RETORT_START_HOUR RETORT_START_MIN START_FILL_HOUR START_FILL_MIN SUPERVISOR_EMP_ID

RFQ	ACCEPT_QTY MATERIAL_ID PRODUCT_ID PROMISE_DATE RECIPE_ID REPLY_PRICE REPLY_QTY REQUESTED_DATE REQUEST_QTY SOLICIT_ID VENDOR_ID
SCHEDULE_D	ACTUAL_PROD_QTY ACTUAL_TIME COMMITED_TIME CUST_LINE_NO ESTIMATED_PROD_QTY MACHINE_ID ORDER_NO SCHEDULE_ID SCHEDULE_SEQ
SCHEDULE_M	BEGIN_PERIOD EMPLOYEE_ID END_PERIOD LAST_REVISED ORIGINAL_DATE SCHEDULE_ID
SHIPMENT	SHIPMENT_DATE SHIPMENT_NO SHIPMENT_TRUCK
SHIP_CROSS_REF	ORDER_NO ORDER_NO_LINE PALLET_ID SHIPMENT_NO
SKILL	HI_WAGE LO_WAGE SKILL_DESC SKILL_ID

SOLICIT	SOLICIT_BY
	SOLICIT_DONE
	SOLICIT_DUE
	SOLICIT_DUE_TIME
	SOLICIT_ID
	SOLICIT_ISSUE
	SOLICIT_REQ_NO
	SOLICIT_TYPE
TASK	TASK_DESC
	TASK_ID
TEST	TEST_DESC
	TEST_ID
TEST_SAMPLE	FILLING_LINE
	INSPECTOR_EMP_ID
	PRODUCTION_DATE
	PRODUCT_ID
	RESULT_1
	RESULT_2
	SAMPLE_HOUR
	SAMPLE_MIN
	TEST_ID
VENDOR	VENDOR_CITY
	VENDOR_CONTACT
	VENDOR_COUNTRY
	VENDOR_FAX
	VENDOR_ID
	VENDOR_NAME
	VENDOR_PHONE
	VENDOR_STATE
	VENDOR_STREET
	VENDOR_STREET2
	VENDOR_TYPE
	VENDOR_ZIP

VENDOR_LOT	EMP_ID LOCATION MATERIAL_ID MATERIAL_LOT_NO VENDOR_ID VLOT_NO VLOT_ON_HAND VLOT_RECEIVE VLOT_RECEIVE_QTY VLOT_STATUS
VENDOR_QUOTE	ACCPT_QTY MATERIAL_ID PRODUCT_ID RECIPE_ID SOLICIT_ID VENDOR_ID
VENXREF	MATERIAL_ID VENDOR_ID VEN_X_NUM
WORK_ORDER	DUE_DATE MACHINE_ID REQ_DEP_ID WORK_ID WO_TYPE
WO_PART	MATERIAL_ID REQ_NO TASK_ID WORK_ID
WO_TASK_EMP	EMP_ID HOURS TASK_ID WORK_ID

APPENDIX B

SQL FORMS

Exhibit B.1

* SALES AND CONTRACT MANAGEMENT
** Current Order and Contract Management [A14]
*** Order Entry [A141]

```
                          Customer Orders
Date        Order No    Cust Id   Name                     Customer PO
________    ________    ______    _____________________    _______________

Employee Id ______    ___________________

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
                                  Label Pack Request
Item Product          Desc         ID   Qty   Date    Quantity   Price    Status
____ ______________  ___________  ____  ___  _______  _________  ________  __
```

CUST_ORDER	DETAIL		PRE-INSERT	1		select :cust_request_date - 10 -((reject_rate+rework_rate) * (:cust_qty/ (avg(rate)*7/100))) into effective_due_date from recipe_line r, product p where r.product_id = :product_id and p.product_id = :product_id group by p.product_id, reject_rate, rework_rate
			PRE-UPDATE	1		select :cust_request_date - 10 - ((reject_rate + rework_rate) * (:cust_qty/ (avg(rate)*7/100))) into effective_due_date from recipe_line r, product p where r.product_id = :product_id and p.product_id = r.product_id group by p.product_id, reject_rate, rework_rate
		CUST_QTY	POST-CHANGE	1		select :cust_qty into production_qty from customer_order_d
		PRODUCT_ID	POST-CHANGE	1		select product_name into desc from product where product_id = :product_id
	ORDER	CUST_ID	POST-CHANGE	1	Customer Not In Master File	select cust_name into name from customer where cust_id = :cust_id

* FACTORY FLOOR CONTROL
** Factory Floor Scheduling [A222]
*** Daily Production Schedule
**** Create New or Add Orders to Daily Schedule

```
Aggregate Schedule ID __________   Begin Date ________   End Date ________

    Daily                                                Begining   Ending
    Schedule Id  Created on   Revised on   Revised By    Date       Date
    ___________  __________   __________   __________    ________   ________
---------------------------------------------------------------------------------
    Order No     Line No      Priority  Product          Production Qty
    ________     ________     ___       _____________    ________   ________
  ___________________
---------------------------------------------------------------------------------
    Machine ID Description                          Commited time  Start time
    ______  ___________________________________     ___________    _________
______  ______  ______
```

DAILY	customer_order_d		KEY-NXTBLK	1		#exemacro commit; nxtblk; exeqry;
			POST-QUERY	1		select commited_time into dummy from schedule_d
			PRE-UPDATE	1		insert into schedule_d (schedule_id,machine_id,order_no,cust_line_no, commited_time) select :schedule_m.schedule_id,machine_id,order_no, cust_line_no, commited_time from schedule_d where schedule_id=:a_schedule_id and order_no=:customer_order_d.order_no and cust_line_no=:customer_order_d.cust_line_no
	schedule_d		POST-UPDATE	1		update schedule_d sd set commited_time = (select sd.commited_time - commited_time from schedule_d where schedule_id = :schedule_d.schedule_id and machine_id = :schedule_d.machine_id) where schedule_id = :schedule_m.a_schedule_id and machine_id = :schedule_d.machine_id
		COMMITED_TIME	POST-FIELD	1	You Have Exceeded Capacity Or No Record	select avail_time from machine_avail m, schedule_m sm where sm.schedule_id =:schedule_d.schedule_id and m.machine_id =:schedule_d.machine_id and sm.end_period - sm.begin_period = m.period_length and m.avail_time > (select (nvl(sum(sd.commited_time),0) + :schedule_d.commited_time) from schedule_d sd where sd.schedule_id = :schedule_d.schedule_id and sd.machine_id =:schedule_d.machine_id and sd.order_no != :customer_order_d.order_no and sd.cust_line_no != :customer_order_d.cust_line_no)
		MACHINE_ID	POST-CHANGE	1	Machine Not In Master File	select machine_desc into desc from machine where machine_id=:machine_id
				2	Machine Is Not For Produciton Line; Plea	select machine_id from machine_process where machine_id=:machine_id *and process_id in (select process_id* from line_process where production_line_id=:customer_order_d.production_line_id)

schedule_m	END_PERIOD	KEY-NXTFLD	1	#exemacro nxtblk; exeqry;
	a_schedule_id	POST-CHANGE	1	select begin_period, end_period into a_begin_period, a_end_period from schedule_m where schedule_id = :a_schedule_id

```
* MANUFACTURING PLANNING
  ** Aggregate Production Scheduling
     *** View an Aggregate Schedule [A 221]



    Schedule ID   Created on  Begin      End        Revised on   Revised by
    __________    _________   _________  _________  _________    __________
--------------------------------------------------------------------------------
  Order No                Machine   Description       Commit time  Start time
  _________  -  _________ _______   _________________ __________   ________
__________


DAY_LIST             schedule_d       MACHINE_ID       POST-CHANGE        1   Machine Not In Master File               select machine_desc
                                                                                                                        into desc
                                                                                                                        from machine
                                                                                                                        where machine_id=:machine_id

                     schedule_m       SCHEDULE_ID      KEY-NXTFLD         1                                             #exemacro nxtblk; exeqry;
```

* FINISHED GOODS CONTROL
** Quality Management [A 41]
*** Incubation Record [A 411]

-------- INCUBATION --------

Production Date	Filling ID	Retort ID	Cook No.
________	_	_	___

No of Samples ____ Disposition _

Incubation Start Date ________ Lead Emp Id ____

Incubation End Date ________ Inspector ID ____

```
* MANUFACTURING PLANNING
  ** Purchasing
     *** Receiving Order

                    Incoming Material

+------------------------------------------------------------------+
|  Material Lot No ________                                        |
|                                                                  |
|  Material Id  Description                      Unit of Measure   |
|  ________     ____________________________        _____          |
|                                                                  |
|  Vendor Id    Name                                               |
|  ________     _____________________________                      |
|                                                                  |
|  Vendor Lot No Received Date Received Qty  Status       Emp Id   |
|  ________      ________      ________      ________     ______   |
+------------------------------------------------------------------+


                 ________

ATL_LOT         VENDOR_LOT      MATERIAL_ID     POST-CHANGE     1  Material Not In Master File     select material_desc,material_uom
                                                                                                    into desc,uom
                                                                                                    from material
                                                                                                    where material_id=:material_id

                                VENDOR_ID       POST-CHANGE     1  Vendor Not In Master File       select vendor_name
                                                                                                    into name
                                                                                                    from vendor
                                                                                                    where vendor_id=:vendor_id
```

* MANUFACTURING PLANNING
** Aggregate Production Scheduling [A221]
*** Schedule Machines [A2214]

```
+-------------------------------------------------------------------------------+
|                                                                               |
| Schedule ID   Begin       End         Created     Last        Revised         |
|               Effective   Effective   On          Revised     By              |
| ___________   _________   _________   _________   _________   _________       |
+-------------------------------------------------------------------------------+
                               Prioritized Orders
Priority Order no  _________     Product           Prod Qty     Prod Line Status
 ____   _________ - _________  ______________     _________     _______    __

                                 MACHINE
      Process    Process Description   Machine   Machine description
      _______    ___________________   _______   ___________________________
```

MC_SCHED	customer_order_d		KEY-LISTVAL	1 2 3 4	#copy :customer_order_d.recipe_id global.recipe #copy :customer_order_d.product_id global.product #exemacro callqry pline_query; #copy global.line customer_order_d.production_line_id
	machine_PROCESS		KEY-LISTVAL	1 2 3 4 5 6 7 8 9	#copy :schedule_m.schedule_id global.schedule #copy :machine_process.machine_id global.machine #copy :customer_order_d.order_no global.order #copy :customer_order_d.cust_line_no global.cust_line #copy :customer_order_d.production_qty global.qty #copy :customer_order_d.production_line_id global.prod_line #copy :customer_order_d.product_id global.product #copy :customer_order_d.recipe_Id global.recipe #exemacro call sched_d;
		MACHINE_ID	POST-CHANGE	1	select machine_desc into desc from machine where machine_id=:machine_id
		process_id	POST-CHANGE	1	select process_desc into pdesc from process where process_id=:process_id

* FACTORY FLOOR CONTROL
** Raw Material Control (A31)
*** Material Move Schedule and Report

```
                 Move Record

 +--------------------------------------+
 |       Product Id ________________    |
 |                                      |
 |     Filling line __________          |
 |                                      |
 |  Material Lot No __________          |
 |                                      |
 |         Move Qty __________          |
 |                                      |
 |  Production Date _________           |
 |                                      |
 |    Move Location _______________     |
 |                                      |
 | Transaction Type _______________     |
 +--------------------------------------+
```

use + qty for moving out of inventory, - for moving back into inventory

MM2	material_move		POST-INSERT	1	*Exceeded Amount On Hand Or Material Lot*	update vendor_lot set vlot_on_hand = vlot_on_hand - :move_qty where material_lot_no = :material_lot_no and vlot_on_hand >= :move_qty
			PRE-DELETE	1		update vendor_lot set vlot_on_hand = vlot_on_hand + :move_qty where material_lot_no = :material_lot_no
			PRE-UPDATE	1	Exceeded On Hand Quantity Of This Vendor	select 'nothing' from material_move mm, vendor_lot vl where vlot_on_hand >= -(move_qty - :move_qty) and mm.product_id=:product_id and mm.material_lot_no = :material_lot_no and mm.production_date=:production_date and mm.transaction_type =:transaction_type and vl.material_lot_no = mm.material_lot_no
				2	Update Failed	update vendor_lot set vlot_on_hand = (select vlot_on_hand-(mm.move_qty - :move_qty) from material_move mm where mm.product_id=:product_id and mm.material_lot_no = :material_lot_no and mm.production_date=:production_date and mm.transaction_type =:transaction_type) where vendor_lot.material_lot_no = :material_lot_no
		FILLING_LINE	POST-CHANGE	1		select 'nothing' from machine where machine_id = :filling_line
		MATERIAL_LOT_NO	POST-CHANGE	1	*Not A Valid Vendor Lot*	*select 'nothing'* from vendor_lot where material_lot_no = :material_lot_no
		PRODUCT_ID	POST-CHANGE	1		select 'nothing' from product where product_id = :product_id

* FACTORY FLOOR CONTROL
** Factory Floor Scheduling [A222]
*** Enter Material Move

```
Schedule ______ Order no ______ Line no ______ Product ______________ Prod Qty ______ Qty to base Move on ______

                    Machine id ______ ______________________________
---------------------------------------------------------------------
    Material ID  Description           Amount to Move   Unit of Measure
    ________     ______________        ____________     ______
                   ______________             ______________

---------------------------------------------------------------------
Material    Lot #     Status    Location   Rec. on   Qty on hand
________    ______    ______    ______     ______    ______  ______
```

MOVE2	recipe		KEY-NXTBLK	1	#exemacro nxtblk; exeqry;
		MATERIAL_ID	POST-CHANGE	1	select material_desc, amt_by_unit*:schedule_d.base_qty,material_uom into matl_desc,move_qty,uom from material m, material_list ml where m.material_id=:recipe.material_id and ml.material_Id = :recipe.material_id and ml.product_Id=:recipe.product_id
	schedule_d		KEY-NXTBLK	1	#exemacro nxtblk; exeqry;
		CUST_LINE_NO	POST-CHANGE	1	select product_name,production_qty,recipe_id,p.product_id into desc, prod_qty, trecipe_id,tproduct_Id from product p, customer_order_d cd where cd.order_no=:order_no and cd.cust_line_no=:cust_line_no and p.product_id=cd.product_id
				2	select begin_period into begin_period from schedule_m where schedule_id=:schedule_id
		MACHINE_ID	POST-CHANGE	1	select machine_desc into mach_desc from machIne where machine_id=:machine_id
	vendor_lot		KEY-COMMIT	1 2 3 4 5	#copy schedule_d.tproduct_id global.product_id #copy schedule_d.machine_Id global.machine_Id #copy vendor_lot.material_lot_no global.material_lot_no #copy schedule_d.begin_period global.start_date #exemacro call move_out; exeqry;

```
* FACTORY FLOOR CONTROL
  ** Factory Floor Reporting [A223]
     *** Pallet Record [A334, A336]
```

```
+------------------------------------------------------------------------+
|                          Pallet Id       Status                        |
|                          _________       ___                           |
|Product  ___________  ____________________________                      |
|  Label  ___________  _________________________  Location  ___________  |
|      Original Order No.       Order No.                                 |
|      ________-________        ________-________                         |
+------------------------------------------------------------------------+
+------------------------------------------------------------------------+
|    Production                                                          |
|    Date       Filler  Retort  Cook No  Label Date  Qty Cases           |
|    _______    __      __      ___      ________    ________            |
|    _______                                                             |
|                                                                        |
+------------------------------------------------------------------------+
```

'ALLET	pallet_card	LABEL_ID	POST-CHANGE	1	Label Id Not In Master File	select label_name into label_desc from label where label_id=:label_id
		PRODUCT_ID	POST-CHANGE	1	Product Not In Master File	select product_name into desc from product where product_id=:product_id

* MANUFACTURING PLANNING
  ** Aggregate Production Scheduling [A221]
    *** Prioritize Orders [A2211]

------ PRODUCT NET REQUIREMENTS ------

```
Due Date   Order No.   Line No.   Customer        Product        Priority  Recipe
________   _________   ________   ____________    _____________  ___       _________
  _                                 ____________            _________
```

```
PRIORITY            customer_order_d   ORDER_NO      POST-CHANGE      1                    select cust_name
                                                                                           into name from customer,customer_order,customer_order_d
                                                                                           where customer_order.order_no=:customer_order_d.order_no
                                                                                           and customer_order.cust_id = customer.cust_id

                                       PRIORITY      POST-CHANGE      1                    select distinct sysdate
                                                                                           into customer_order_d.last_prioritized_date
                                                                                           from customer

                                       recipe_id     POST-CHANGE      1   Not In Recipe    select recipe_id
                                                                                           into recipe_id
                                                                                           from recipe
                                                                                           where recipe_id=:customer_order_d.recipe_id
                                                                                           and product_id=:customer_order_d.product_id
```

* SALES AND CONTRACT MANAGEMENT
** Contract Pricing [A13]
*** Process Plan [A1322]

-------- PROD_LINE --------

Line Id	Description

-------- LINE_PROCESS --------

Process Id	Process Description	Sequence

-------- MACHINE_PROCESS --------

Machine ID	Description

ROD_LINE	line_process	PROCESS_ID	POST-CHANGE	1	select process_desc into descr from process where process_id = :line_process.process_id
	machine_process	MACHINE_ID	POST-CHANGE	1	select machine_desc into desc from machine where machine_id =:machine_process.machine_id

```
* SALES AND CONTRACT MANAGEMENT
  ** Provide Quotation [A 11]
     *** Pricing Information [ A 111]


                Product Price

          Product Id         Product Name
          _______________    ______________________________

Net Wt _________   Can Size _______________   Can Spec _____   Lid Spec _____

Qty/Case _________   Std Reject Rate _________   Prices Valid until _________

+----------------------------------------------------------------------------+
+----------------------------------------------------------------------------+
                    Break Quantity      Price
_______________     __________      __________
```

* MANUFACTURING PLANNING
** Purchasing
*** Purchase Orders [A312]

```
PO Number _______            Requisition ID        CRAMTD PURCHASE ORDER
  PO Date ________           __________            ---------------------
PO Status __                                        RUTGERS UNIVERSITY
                                                    CAFT CENTER
                                                    COLLEGE FARM ROAD
     Vendor ID __________                           NEW BRUNSWICK, NJ  08903
   Vendor Name __________________________
Vendor Address __________________________
               ________________ __
               __________
--------------------------------------------------------------------------------
_______ Material  Material                  Quantity  Unit  Total  Promised  Line
Line No.   ID     Description          UOM  Ordered   Cost  Cost     Date   Status
```

_ORDER	purchase_order	REQ_ID	POST-CHANGE	1	Requisition Id Not In Master File	select req_id from requisition where requisition.req_id=:purchase_order.req_id
		VENDOR_ID	POST-CHANGE	1	Vendor Id Not In Master File	select vendor_name into purchase_order.v_name from vendor where vendor.vendor_id=:purchase_order.vendor_id
				2		select vendor_street into purchase_order.v_street from vendorwhere vendor.vendor_id=:purchase_order.vendor_id
				3		select vendor_city into purchase_ord from vendorwhere vendor.vendor_id=:purchase_order.vendor_id
				4		select vendor_state into purchase_order.v_state from vendor where vendor.vendor_id=:purchase_order.vendor_id
				5		select vendor_zip into purchase_order.v_zip from vendor where vendor.vendor_id=:purchase_order.vendor_id
	purchase_order_item	MATERIAL_ID	POST-CHANGE	1	Material Id Not In Master File	select material_desc into purchase_order_item.mat_desc from material where material.material_id=:purchase_order_item.material_id
				2		select material_uom into purchase_order_item.mat_uom from material where material.material_id=:purchase_order_item.material_id
		PO_ITEM_PRICE	POST-CHANGE	1		select :purchase_order_item.po_item_price*:purchase_order_item.po_item_qty into :purchase_order_item.tot_cost from po_item

* FACTORY FLOOR CONTROL
** Quality Control
*** Incoming Material Acceptance Report [A3133, A3136]

```
+-------------------------------Vendor Lot-------------------------------+
|             Material Lot Number ________                               |
|  Vendor ID ________                           Vendor Lot ___________  |
| Vendor Name _____________                  Received Date _______     |
| Material ID ________              Quantity Received (lbs) ___________  |
| Description ________________      Quantity on Hand (lbs) ________     |
|                 Inspector ID ____                  Status __           |
+------------------------------------------------------------------------+
|                     Test ID ________                                   |
| ________                                                               |
| ________       Description ________________________                    |
|                 Test Result _________                                  |
+------------------------------------------------------------------------+
```

QUALITY_REPORT	QUALITY	TEST_ID	POST-CHANGE	1	Test Id Not In Master File.	select test_desc into test_name from test where test.test_id=:quality.test_id
	QUALITY_REPORT	TEST_ID	POST-CHANGE	1	Test Not In Master File	select test_desc into desc from test where test_id=:test_id
	VENDOR_LOT	MATERIAL_ID	POST-CHANGE	1		select material_desc select material_desc into material_desc into material_desc from material from material where material.material_id=:vendor_lot.material_id where material.material_id=:vendor_lot.material_id
		Material_desc	POST-CHANGE	1	Material Id Not In Master File	select material_desc select material_desc into vendor_lot.material_desc into vendor_lot.material_desc from material from material where material.material_id=:vendor_lot.material_id where material.material_id=:vendor_lot.material_id
		VENDOR_ID	POST-CHANGE	1	Vendor Id Not In Master File	select vendor_name select vendor_name into vendor_lot.ven_name into vendor_lot.ven_name from vendor from vendor where vendor.vendor_id=:vendor_lot.vendor_id where vendor.vendor_id=:vendor_lot.vendor_id

```
* SALES AND CONTRACT MANAGEMENT
  ** Provide Quotation [A11]
     *** Quotation [A113]

                               Quotation

Quotation Id    ________       Customer Id     ______

Quotation Date  ________       Customer Name   __________________________

Valid Until     ________

--------------------------------------------------------------------------------

Line                                              Requested  Unit    Total
Item  Product          Description      Quantity  Delivery   Price   Price
____  _______________  _______________  ________  _________  ______  _______

QUOTE                 quote            CUST_ID        POST-CHANGE     1   Customer Not In Master File     select cust_name
                                                                                                          into name
                                                                                                          from customer
                                                                                                          where cust_id=:cust_id

                      quote_d          PRODUCT_ID     POST-CHANGE     1                                   select product_name
                                                                                                          into desc
                                                                                                          from product
                                                                                                          where product_id=:product_id

                                                                      2                                   select least(valid_till,:quote.experation_date)
                                                                                                          into quote.experation_date
                                                                                                          from product
                                                                                                          where product_id=:product_id

                                       QUOTE_PRICE    POST-CHANGE     1                                   select :quote_price * :quote_qty
                                                                                                          into total_price
                                                                                                          from quote

                                       QUOTE_QTY      POST-CHANGE     1                                   select min(break_price)
                                                                                                          into quote_price
                                                                                                          from price_break  pb
                                                                                                          where pb.product_id=:product_id
                                                                                                          and pb.break_qty <= :quote_qty
```

* FACTORY FLOOR CONTROL
  ** Raw Material Control [A31]
    *** Receiving Report [A312, A3131]

```
                 -------- RECEIVING REPORT --------
     P.O. NUMBER ______          RECEIVED DATE ________
                 ________  P.O. STATUS __
------------------------------------------------------------------------
  P.O. LINE NUMBER ______
       MATERIAL ID ________             RECEIVED QTY ________
     MATERIAL DESC ____________        VENDOR LOT NO ________
P.O. ORIGINAL QTY ________            PO ITEM STATUS _
  P.O. BALANCE QTY ________
------------------------------------------------------------------------
  GIVEN THE PREVIOUS DATA A MATERIAL LOT WILL BE CREATED WITH:
  MATERIAL LOT NO ________                 VENDOR ID ________
      MATERIAL ID ________             VENDOR LOT NO ________
                     QUANTITY RECEIVED ________
                         DATE RECEIVED _______
------------------------------------------------------------------------

  MATERIAL_LOT_NO ________                 PO_NO ______
          PO_ITEM ________
                                                 _
```

RECEIVING_REPORT	INVOICE_PAY	PO_NO	PRE-FIELD	1		select max(to_number(vl.material_lot_no)) into :invoice_pay.material_lot_no from vendor_lot vl
	PO_ITEM	MATERIAL_ID	POST-CHANGE	1		select m.material_desc into :po_item.material_desc from material m where m.material_id = :po_item.material_id
		PO_ITEM	POST-CHANGE	1		select po.po_item_qty, po.po_item_bal into :po_item.po_Item_qty, :po_item.po_item_bal from po_Item po where po.po_no = :po_item.po_no and po.po_item = :po_item.po_item
		VLOT_RECEIVE_ QTY	POST-CHANGE	1	Received Quantity Entered Exceeds The P.	select po.po_item_bal - :po_item.vlot_receive_qty into :po_item.po_Item_bal from po_Item po where po.po_no = :po_item.po_no and po.po_item = :po_item.po_item and po.po_item_bal >= :po_item.vlot_receive_qty
	PURCHASE_ORDER	PO_NO	POST-CHANGE	1	Purchace Order Does Not Exist	select pu.vendor_id into :purchase_order.vendor_id from purchase_order pu where pu.po_no = :purchase_order.po_no
	VENDOR_LOT	MATERIAL_ID	POST-CHANGE	1		select max(to_number(vl.material_lot_no)) + 1 into :vendor_lot.material_lot_no from vendor_lot vl

* FINISHED GOODS CONTROL
** Lot Tracking/ Tracebility
*** Lot Tracking (Finished Goods -> Raw)

```
               -------- RECEIVING REPORT --------
     P.O. NUMBER ______           RECEIVED DATE ________
                 _________   P.O. STATUS _
-----------------------------------------------------------------------------
  P.O. LINE NUMBER ______
       MATERIAL ID _________          RECEIVED QTY _________
     MATERIAL DESC _____________     VENDOR LOT NO _________
  P.O. ORIGINAL QTY _________        PO ITEM STATUS _
  P.O. BALANCE QTY _________
-----------------------------------------------------------------------------
   GIVEN THE PREVIOUS DATA A MATERIAL LOT WILL BE CREATED WITH:
   MATERIAL LOT NO _________               VENDOR ID _________
       MATERIAL ID _________           VENDOR LOT NO _________
                  QUANTITY RECEIVED _________
          ______       DATE RECEIVED _______
-----------------------------------------------------------------------------

   MATERIAL_LOT_NO _________                 PO_NO ______
           PO_ITEM _________
                                                   _
```

REPORT	RETORT	HOUR_RETORT_ START	POST-CHANGE	1	select trunc((hour_retort_start+min_retort_start/60)-.75),round((round ((hour_retort_start+min_retort_start/60)-.75,2)-trunc((hour_retort_start+ min_retort_start/60)-.75))*60) into hour,min from retort where cook_no=:cook_no and production_date=:retort.production_date
		RETORT_START_ Hour	POST-CHANGE	1	select trunc((hour_retort_start+min_retort_start/60)-.75),round((round ((hour_retort_start+min_retort_start/60)-.75,2)-trunc((hour_retort_start+ min_retort_start/60)-.75))*60) into hour,min from retort where cook_no=:cook_no and production_date=:retort.production_date
		product_id	POST-CHANGE	1	select product_name into desc from product where product_id=:retort.product_id
	batch_d	MATERIAL_ID	POST-CHANGE	1	select material_desc into description from material where material_id=:material_id
		VENDOR_ID	POST-CHANGE	1	select vendor_name into name from vendor where vendor_id=:vendor_id
		material_lot_ no	POST-CHANGE	1	select m.material_id, m.material_desc into mat_id, desc from vendor_lot vl, material m where vl.material_lot_no =:material_lot_no and m.material_id = vl.material_id

```
+------------------------------------------------------------------+
|                        Requisition Id                            |
|                        ______________                            |
|                                                                  |
|   Material Id Desc                 Qty        Units  Req Date  Dept |
|   ___________ ____________________ __________ _____ _________ ______ |
|   ____                                                           |
+------------------------------------------------------------------+
```

ɪEQ2	requisition		POST-INSERT	1		update req_seq set requisition_no = requisition_no + 1 where dummy_key ='100'
		MATERIAL_ID	POST-CHANGE	1	Material Not In Master File	select material_desc,material_uom into desc,unit_of_measure from material where material_id=:material_id
				2		select sysdate,requisition_no,'op' into req_date, req_id, req_status from req_seq where dummy_key='100'

```
* FACTORY FLOOR CONTROL
  ** Quality Control
     *** Retort Record

                              Actual
-----------------------Retort-Information---------------------------
   Product _______________   Description ____________________

       Production   Filling  Retort  Cook  Start      End
        date        Line     Id      No    Fill time  Fill time
       ________     _        _       ____  __:__      __:__

Retort Start Time __:__          Initial Temp _________
    End Vent Time __:__         End Vent Temp _________
 End Come-up Time __:__      End Come-up Temp _________
    End Cook Time __:__         End Cook Temp _________
    Supervisor Id ____        Retort Quantity _________
```

* MANUFACTURING PLANNING
** Aggregate Production Scheduling
*** Schedule Machines [A2214]

```
------ SCHEDULING OF  MATERIALS ------

    Priority  Order No.     Product          Prod Qty   Status
    ________  _________  _  ______________  _________  ______

________ -------- MATERIAL REQUIREMENTS -------- ________

Material Id  Description   Net required  Net available  Order Amt.  Units
___________  ___________  ____________  _____________  __________  _____
```

```
SCHEDULE  customer_order_d  ORDER_NO     POST-CHANGE  1   select ceil(:production_qty/amt_end_units)
                                                          into temp
                                                          from recipe_master
                                                          where product_id =:customer_order_d.product_id
                                                          and recipe_id =:customer_order_d.recipe_id

          recipe                         KEY-COMMIT   1   #copy recipe.material_id global.material
                                                      2   #copy recipe.unit global.unit
                                                      3   #copy recipe.desc global.desc
                                                      4   #copy recipe.avail_req global.amt
                                                      5   #exemacro call requisition;

                            MATERIAL_ID  POST-CHANGE  1   select material_desc
                                                          into :desc
                                                          from material
                                                          where material_id = :recipe.material_id

                                                      2   select unit_of_measure
                                                          into :unit
                                                          from material_list
                                                          where material_id=:recipe.material_id
                                                          and product_id=:recipe.product_id

                                                      3   select (amt_by_unit*:customer_order_d.temp)/(recovery_percent/100)
                                                          into :required
                                                          from material_list
                                                          where product_id=:recipe.product_id
                                                          and material_id=:recipe.material_id

                                                      4   select nvl(sum( ceil(c.production_qty/rm.amt_end_units)*
                                                          m.amt_by_unit/(m.recovery_percent/100) ),0)  into :commited
                                                          from customer_order_d c, recipe_master rm, material_list m
                                                          where m.material_id=:recipe.material_id
                                                          and c.cust_line_status='sc' and c.production_qty > 0
                                                          and rm.product_id = c.product_id
                                                          and rm.recipe_id = c.recipe_id
                                                          and m.material_id in (
                                                            select material_id from recipe r
                                                              where r.product_id = c.product_id
                                                              and r.recipe_id = c.recipe_id)

                                                      5   select nvl(sum(v.vlot_on_hand),0)
                                                          into :tv_lot
                                                          from vendor_lot v
                                                          where :recipe.material_id=v.material_id

                                                      6   select nvl(sum(p.po_item_bal),0)
                                                          into :tpo_bal
                                                          from po_item p
                                                          where :recipe.material_id=p.material_id
```

7

```
select nvl(sum(r.req_qty),0)
into :tr_qty
from requisition r
where :recipe.material_id=r.material_id and
      r.req_status = 'op'
```

8

```
select (:tv_lot + :tpo_bal + :tr_qty - :commited)
into :avail
from recipe
```

9

```
select greatest(0,(:recipe.required - :recipe.avail))
into :avail_req
from material
where material_id=:recipe.material_id
```

```
* FACTORY FLOOR CONTROL
  ** Quality Control
     *** Test Material

   +------------------------------------------+
   |       Test ID     Description            |
   |       _____       ___________________    |
   +------------------------------------------+

      Material Id  Description          Low limit  High limit
      _______      _________________    ______     ______
  ____


TEST_M         material_spec    MATERIAL_ID    post_change    1        select material_desc
                                                                       into desc
                                                                       from material
                                                                       where material_id=:material_id
```

```
* FACTORY FLOOR CONTROL
  ** Quality Control
     *** Seal Strength / Residual Gas Report

+-----------------------------------------------------------------+
|                  -------- TEST_SAMPLE --------                  |
| Production                                    Filling  Sample  |
|    Date     Product              Description   Line     Time   |
| ________    ______________       ___________    _       __:__  |
|                                                                 |
|                 Test ID          Test Description                |
|                 ______           __________________              |
|                                                                 |
|   Pouch 1 Result ________            Inspector ID ______        |
|   Pouch 2 Result ________                                        |
+-----------------------------------------------------------------+
```

TEST_SAMPLE	Test_Sample	TEST_ID	POST-CHANGE	1	Test Not In Master File.	select test_desc into test_sample.test_description from test where test.test_id=:test_sample.test_id
		product_id	POST-CHANGE	1	Product Not In Master File	select product_name into desc from product where product_id =:product_id

* FACTORY FLOOR CONTROL
** Factory Floor Scheduling [A222]
*** Daily Production Schedule [A2221]
**** Make changes to a Daily Schedule

```
Aggregate Schedule ID ________   Begin Date ________   End Date ________

    Daily                                                Begining   Ending
    Schedule Id  Created on  Revised on  Revised By      Date       Date
    ________     ________    ________    ________        ________   ________
-------------------------------------------------------------------------------
    Order No     Line No     Priority    Product         Production Qty
    ________     ________    ____        ____________    ________
-------------------------------------------------------------------------------
    Machine ID Description                        Commited time   Start time
    ______     ____________________________       ________        ________
```

JP_DAY	customer_order_d		KEY-NXTBLK	1		#exemacro nxtblk; exeqry;
	schedule_d		PRE-DELETE	1		update schedule_d sd set commited_time = commited_time + :commited_time where schedule_id=:schedule_m.a_schedule_id and machine_id =:schedule_d.machine_id
			PRE-UPDATE	1		update schedule_d sd set commited_time = (select sd.commited_time -(:commited_time-commited_time) from schedule_d where schedule_id = :schedule_d.schedule_id and machine_id = :schedule_d.machine_id) where schedule_id = :schedule_m.a_schedule_id and machine_id = :schedule_d.machine_id
		COMMITED_TIME	POST-FIELD	1	You Have Exceeded Capacity Or No Record	select avail_time from machine_avail m, schedule_m sm where sm.schedule_id =:schedule_d.schedule_id and m.machine_id =:schedule_d.machine_id and sm.end_period - sm.begin_period = m.period_length and m.avail_time > (select (nvl(sum(sd.commited_time),0) + :schedule_d.commited_time) from schedule_d sd where sd.schedule_id = :schedule_d.schedule_id and sd.machine_id =:schedule_d.machine_id and sd.order_no != :customer_order_d.order_no and sd.cust_line_no != :customer_order_d.cust_line_no)
		MACHINE_ID	POST-CHANGE	1	Machine Not In Master File	select machine_desc into desc from machine where machine_id=:machine_id
				2	Machine Is Not For This Produciton Line;	select machine_id from machine_process where machine_id=:machine_id and process_id in (select process_id from line_process where production_line_id=:customer_order_d.production_line_id)
	schedule_m	END_PERIOD	KEY-NXTFLD	1		#exemacro nxtblk; exeqry;
		a_schedule_id	POST-CHANGE	1		select begin_period, end_period into a_begin_period, a_end_period from schedule_m where schedule_id = :a_schedule_id

```
* MANUFACTURING PLANNING
  ** Purchasing
     *** Vendor Information [A312]


                   Vendor Master File
+-----------------------------------------------------+
| Vendor Id _____      Name _______________          |
|   Address ____________                              |
|           ____________                              |
|      City _______  State __  Country ___  Zip _____ |
+-----------------------------------------------------+
      Phone _____     Fax _____
Contact Name _______
       Type __
```

* MANUFACTURING PLANNING
  ** Purchasing
    *** Vendor Information [A312]

```
+---------------Vendor---Material-cross-reference-----------------------------+
| Material Id Description      Vendor Id  Name              Cross ref. no.     |
| ___________ ________________ __________ ________________  ___________________|
|                                                                              |
|                                                                              |
+------------------------------------------------------------------------------+
```

VENXREF	venxref	MATERIAL_ID	POST-CHANGE	1	Material Not In Master File	select material_desc into desc from material where material_id=:material_id
		VENDOR_ID	POST-CHANGE	1	Vendor Not In Master File	select vendor_name into name from vendor where vendor_id=:vendor_id

* MANUFACTURING PLANNING
** Aggregate Production Scheduling
*** Committed Finished Goods

```
                 ------ PRIORITIZED CUSTOMER ORDERS ------

Priority   Order No.   Line No.  Customer        Product          Due Date
  ___      _________   ________  ______________  _______________  _________

         Order Quantity  _________   Production Quantity  _________
   __  _____________

                 -------- AVAILABLE PALLETS  --------

 Production date    Quantity of cases      Label Id        Total cases for day
  __________         ___________       _______________       ___________

   If you wish to commit this pallet to the above customer order enter "C"
   To commit the whole day's production enter "D"'
   To see the next available pallet enter "N".
   To go to the next customer order enter "F".

                  Enter choice here ------> _
     ___________  ___________                                  _______________
```

```
VPROD_QTY      PALLET_CARD              DAY            1    select :customer_order_d.order_no,:customer_order_d.cust_line_no
                                                            into pallet_card.order_no,pallet_card.cust_line_no
                                                            from customer

                                                       2    select (:production_qty-:total_cases)
                                                            into customer_order_d.production_qty
                                                            from customer

                                        KEY-NXTFLD     1    #exemacro case choice is
                                                            when 'c' then exetrg pallet; commit;
                                                            when 'n' then nxtrec;
                                                            when 'f' then prvblk; nxtrec; nxtblk; exeqry;
                                                            when 'd' then exetrg day; commit;
                                                            end case;

                                        PALLET         1    select :customer_order_d.order_no, :customer_order_d.cust_line_no
                                                            into pallet_card.order_no,pallet_card.cust_line_no
                                                            from customer

                                                       2    select (:production_qty-:cases)
                                                            into customer_order_d.production_qty
                                                            from customer

                                        POST-UPDATE    1    update pallet_card
                                                            set order_no=:customer_order_d.order_no,
                                                                cust_line_no=:customer_order_d.cust_line_no
                                                                where pallet_id in (select pc.pallet_id from pallet_d pd,pallet_card pc
                                                                   where pc.product_id=:pallet_card.product_id
                                                                   and(pc.label_id='shiner' or pc.label_id=:customer_order_d.label_id)
                                                                   and pd.production_date= to_date(:pdate)
                                                                   and pc.pallet_id=pd.pallet_id
                                                                   and pc.order_no is null))
                                                            and :choice='d'

                            PALLET_ID   POST-CHANGE    1    select qty_cases
                                                            into cases
                                                            from pallet_d
                                                            where pallet_d.pallet_id = :pallet_card.pallet_id

                                                       2    select production_date
                                                            into pdate
                                                            from pallet_d
```

```
                                              3          select sum(qty_cases)
                                                         into total_cases
                                                         from pallet_card pc, pallet_d pd
                                                         where pd.production_date = to_date(:pdate)
                                                         and order_no is null
                                                         and pc.pallet_id=pd.pallet_id
                                                         and pc.product_id=:pallet_card.product_id
                                                         and (pc.label_id='shiner' or
pc.label_id=:customer_order_d.label_id)

           customer_order_d   ORDER_NO   POST-CHANGE  1  select cust_name
                                                         into name from
customer,customer_order,customer_order_d
                                                         where
customer_order.order_no=:customer_order_d.order_no
                                                         and customer_order.cust_id = customer.cust_id
```

* FACTORY FLOOR CONTROL
** Factory Floor Reporting
*** Batch Sheet Report (A322,A336)

=====BATCH SHEET RECORD=====

```
PRODUTION DATE ________                BATCH START __ __
    PRODUCT ID ______________          BATCH SIZE ________
               ______________________
                                        COOK TEMP ________
    KETTLE ID ______
                                        COOK TIME __ __
  FILLING LINE ______
---------------------------------------------------------------------------
  MATERIAL LOT NO ________       MATERIAL ID ______
                                 MATERIAL
        BATCH QTY ________       DESCRIPTION __________________
```

customer_order_d ORDER_NO POST-CHANGE 1

```
where pallet_d.pallet_id=:pallet_id

select sum(qty_cases)
into :total_cases
from pallet_card pc, pallet_d pd
where pd.production_date = to_date(:pdate)
and [illegible] null
and pc.pallet_card.pallet_id
and pc.product_id=:pallet_card.product_id
and (pc.label_id [illegible] or pc.label_id=:customer_order_d.label_id)

select cust_name
into :name from customer,customer_order,customer_order_d
where customer_order.order_no=:customer_order_d.order_no
and customer_order.cust_id = customer.cust_id
```

FACTORY FLOOR CONTROL
** Factory Floor Reporting
*** Batch Sheet Report (A322,A336)

=====BATCH SHEET RECORD=====

RODUTION DATE ________ BATCH START __ __

PRODUCT ID ______________ BATCH SIZE _________

KETTLE ID ______ COOK_TEMP _________

FILLING LINE ______ COOK TIME __ __

MATERIAL LOT NO ________ MATERIAL ID ______

BATCH QTY ________ MATERIAL DESCRIPTION ________________

Form Name: d_bat_d

```
                    ========  BATCH_D  ========

   PRODUCTION_DATE __________            PRODUCT_ID ________________
   MATERIAL_LOT_NO ___________         FILLING_LINE _
 BATCH_START_HOUR ____             BATCH_START_MIN ____
         BATCH_QTY ___________
```

Form Name: d_bat_record

======== BATCH_RECORD ========

PRODUCT_ID ________________	FILLING_LINE ________
PRODUTION_DATE __________	BATCH_START_HOUR ____
BATCH_START_MIN ____	KETTLE_ID ________
BATCH_SIZE ___________	COOK_TEMP ___________
COOK_TIME_HOUR ____	COOK_TIME_MIN ____
SUPERVISOR_EMP_ID _____	RECIPE_ID ___________

Form Name: d_cust_d

```
                 ========  CUST_ORDER_D  ========
         ORDER_NO __________          CUST_LINE_NO __________
       PRODUCT_ID _______________         CUST_QTY __________
 CUST_LINE_STATUS __             CUST_REQUEST_DATE _________
            PRICE __________              LABEL_ID _______________
         PRIORITY ____          LAST_PRIORITIZED_D _________
        RECIPE_ID __________        PRODUCTION_QTY __________
      PACKING_QTY __________    PRODUCTION_LINE_ID _______
            DUMMY __________    EFFECTIVE_DUE_DATE _________
```

Form Name: d_filling_record

======== FILLING_RECORD ========

PRODUCT_ID ____________ PRODUTION_DATE __________

FILLING_LINE ______ QTY_FILLED __________

MATERIAL_LOT_NO __________

Form Name: d_label

======== LABEL ========

LABEL_ID ________________

LABEL_NAME ________________________________

Form Name: d_material_list

======== MATERIAL_LIST ========

PRODUCT_ID ______________ MATERIAL_ID __________

UNIT_OF_MEASURE _______ RECOVERY_PERCENT __________

AMT_BY_UNIT __________

Form Name: d_pal_c

======== PALLET_CARD ========

PALLET_ID ______________	LOCATION ____________________
PRODUCT_ID ____________________	LABEL_ID ____________________
ORDER_NO ______________	CUST_LINE_NO ______________
ORIG_ORDER_NO ______________	ORIG_CUST_LINE_NO ______________
PALLET_STATUS _____	VAR_UNITS ______________
VAR_DATE ____________	VAR_STATUS ______

Form Name: d_pal_d

======== PALLET_D ========

PALLET_ID __________	FILLING_LINE _______
PRODUCTION_DATE _________	RETORT_ID _______
COOK_NO ____	LABEL_DATE _________
QTY_CASES __________	

Form Name: d_product

```
                    ========  PRODUCT  ========
     PRODUCT_ID ______________
   PRODUCT_NAME ____________________________
       CAN_SIZE ______________              CAN_SPEC _____
       LID_SPEC _____                     NET_WEIGHT __________
   QTY_PER_CASE __________               REJECT_RATE __________
    VALID_TILL _________                 REWORK_RATE __________
```

Form Name: d-recipe_master

======== RECIPE_MASTER ========

PRODUCT_ID ____________	RECIPE_ID ________
CSIZE ________	GALLONS ________
AMT_END_UNITS ________	TYPE_COOK ____________
PROCESS_TIME ________	PROCESS_TEMP ________
TARGET_INIT_TEMP ________	TARGET_COOK_TIME ________
TARGET_COOK_TEMP ________	

Form Name: d_retort

======== RETORT ========

FILLING_LINE ________	PRODUCTION_DATE ________
RETORT_ID ________	COOK_NO ____
INITIAL_TEMP ________	NO_CARTS ____
RETORT_START_HOUR ________	RETORT_START_MIN ________
END_VENT_HOUR ________	END_VENT_MIN ________
END_VENT_TEMP ________	END_COMEUP_HOUR ________
END_COMEUP_MIN ________	END_COMEUP_TEMP ________
END_COOK_HOUR ________	END_COOK_MIN ________
END_COOK_TEMP ________	SUPERVISOR_EMP_ID _____
RETORT_QTY _____	NO_SAMPLES _____

```
Form Name: d_shipment


                    ========  SHIPMENT  ========
     SHIPMENT_NO __________          SHIPMENT_DATE _________
   SHIPMENT_TRUCK _____
```

Form Name: d_ship_cross_ref

```
========  SHIP_CROSS_REF  ========

PALLET_ID __________          SHIPMENT_NO __________
 ORDER_NO __________        ORDER_NO_LINE _____
```

Form Name: d_vendor_lot

======== VENDOR_LOT ========

MATERIAL_LOT_NO	__________	VENDOR_ID	__________
MATERIAL_ID	__________	VLOT_NO	__________
VLOT_RECEIVE	_________	VLOT_STATUS	__________
VLOT_ON_HAND	__________	VLOT_RECEIVE_QTY	__________
EMP_ID	_______	LOCATION	______

Form Name: d_invoice

======== INVOICE_PAY ========

MATERIAL_LOT_NO ____________	PO_NO ________
PO_ITEM ________	INVOICE_QTY ____________
INVOICE_NO __________________	

Form Name: d_material

======== MATERIAL ========

```
     MATERIAL_ID __________
   MATERIAL_DESC ________________________________________
    MATERIAL_UOM _____                     REORDER_POINT __________
     REORDER_QTY __________                    LAST_PAID __________
        LAST_BUY _________                      STD_COST __________
```

Form Name: d_recipe

======== RECIPE ========

PRODUCT_ID ________________ RECIPE_ID ___________

MATERIAL_ID ___________

Form Name: d_machine

======== MACHINE ========

MACHINE_ID ________

MACHINE_DESC ________________________________

MACHINE_LABOR ___________ MACHINE_LOCATION ________________

APPENDIX C

SQL REPORTS

* SALES AND CONTRACT MANAGEMENT
** Provide Quotation (A11)
*** Current Material Pricing

Report Name: mat_pricing

CURRENT MATERIAL PRICING PER CONTAINER: STANDARD vs. ACTUAL

Product ID: 1137   Product Name: MINESTRONE

		Unit Price ($/Unit)			Container Cost ($/Container)	
		At Std	At Actual	Date Last Actual	At Std	At Actual
Material ID:	61210	0.65	0.58	05-AUG-91	0.036	0.032
Material Desc:	BEANS, CICI					
Unit of Measure:	LBS					
Qty per Container:	0.0498					
Recovery Percent:	90.0%					
Gross Qty per Cont:	0.0553					

Query Settings

Query Name: mat   Query 1 of 1

```
select p.product_id, p.product_name, m.material_id, m.material_desc,
       ml.unit_of_measure, (ml.amt_by_unit/(rm.amt_end_units*p.qty_per_cas
        ml.recovery_percent, ((ml.amt_by_unit/(p.qty_per_case*rm.amt_end_un
        m.std_cost, m.last_paid, m.last_buy,
        (((ml.amt_by_unit/(rm.amt_end_units*p.qty_per_case))/(ml.recovery_p
        (((ml.amt_by_unit/(p.qty_per_case*rm.amt_end_units))/(ml.recovery_p
from material m, material_list ml, product p, recipe_master rm
where p.product_id=:product__id and p.product_id=ml.product_id and
```

Summary Settings 1 of 2

Summary Name	Field	Function	Data Type	Width	Display Format
S_SUM	S_TOT	Sum	NUM	6	$BZZ9.99
A_SUM	A_TOT	Sum	NUM	6	$BZZ9.99

Summary Settings 2 of 2

Summary Name	Print Group	Reset Group
S_SUM	G_info	G_info
A_SUM	G_info	G_info

* SALES AND CONTRACT MANAGEMENT
** Order/ Contract Management (A14)
*** Customer Order Status (A143)

Report Name: cust5

CUSTOMER ORDER STATUS REPORT

Customer Order No: 123456 Customer Name: Campers International
Customer PO Number: 9ADFD-3

Line Item No: 1
Product ID: 1026
Product Name: CHICKEN BROTH

Order Quantity: 8
Delivery Date: 10-JUN-91
Open/Sched Qty:
-64
928

Schedule Quantity	Inventory		Shipped	
	Units	Prod Date	Units	Ship Date
	36	02-MAR-91		
	36	03-MAR-91		Total
	72	Total		

Query Settings

Query Name: cus Query 1 of 5

```
select cd.crder_no, co.cust_po, c.cust_name, cd.cust_line_no, cd.product_i
       p.product_name, cd.cust_qty, cd.cust_request_date
 from   customer_order_d cd, customer_order co, customer c, product p
 where  ((co.cust_po=:Customer_PO_Number and c.cust_name=:Customer_Name) or
        cd.order_no=:Order_Number) and cd.order_no=co.order_no and
        co.cust_id=c.cust_id and cd.product_id=p.product_id
order by cd.cust_line_no
```

Query Name: pal   Query 2 of 5

```
select pc.order_no, pc.cust_line_no, (cd.cust_qty - sum(pd.qty_cases))
from customer_order_d cd, pallet_card pc, pallet_d pd
 where pd.pallet_id=pc.pallet_id and pc.order_no=cd.order_no and
       pc.cust_line_no=cd.cust_line_no
 group by pc.order_no, pc.cust_line_no, cd.cust_qty
```

Parent Query 1: cus   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
CUST_LINE_NO	CUST_LINE_NO	

Query Name: sched   Query 3 of 5

```
select order_no, cust_line_no, production_qty
from customer_order_d cd
 where order_no IN (select order_no from schedule_d
                   where order_no=cd.order_no
and cust_line_no = cd.cust_line_no)
```

Parent Query 1: cus   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
CUST_LINE_NO	CUST_LINE_NO	

Query Name: inv   Query 4 of 5

```
select pc.order_no, pc.cust_line_no, sum(pd.qty_cases), pd.production_date
from pallet_card pc, pallet_d pd
 where pc.pallet_id  NOT IN (select sxr.pallet_id from ship_cross_ref sxr)
       and pc.pallet_id=pd.pallet_id
 group by pc.order_no, pc.cust_line_no, pd.production_date
order by pd.production_date
```

Parent Query 1: cus   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
CUST_LINE_NO	CUST_LINE_NO	

Query Name: shp   Query 5 of 5

```
select sxr.order_no, sxr.order_no_line, s.shipment_date, sum(pd.qty_cases)
from   pallet_d pd, ship_cross_ref sxr, shipment s
 where  sxr.pallet_id=pd.pallet_id and sxr.shipment_no=s.shipment_no
 group by sxr.order_no, sxr.order_no_line, s.shipment_date
order by s.shipment_date
```

Parent Query 1: cus   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
ORDER_NO_LINE	CUST_LINE_NO	

Summary Settings 1 of 2

Summary Name	Field	Function	Data Type	Width	Display Format
INV_SUM	INVEN	Sum	NUM	6	
SHP_SUM	SHIP	Sum	NUM	6	

Summary Settings 2 of 2

Summary Name	Print Group	Reset Group
INV_SUM	G_opn	G_opn
SHP_SUM	G_opn	G_opn

* SALES AND CONTRACT MANAGEMENT
  ** Current Order and Contract Management (A14)
    *** Contract Status (A141)

Report Name: con_stat

DSPC CONTRACT STATUS REPORT

Customer Order No: 123456 Customer Name: Campers International
Customer PO Number: 9ADFD-3

Line Item No: 1
Product ID: 1026
Product Name: CHICKEN BROTH

Order Quantity: 8
Delivery Date: 10-JUN-91
Open/Sched Qty:

Built, Non-Compliance		Built, In Compliance		Shipped	
Units	Prod Date	Units	Prod Date	Units	Ship Date
	Total		Total		Total

Query Settings

Query Name: cus Query 1 of 5

```
select cd.order_no, co.cust_po, c.cust_name, cd.cust_line_no, cd.product_i
        p.product_name, cd.cust_qty, cd.cust_request_date
  from   customer_order_d cd, customer_order co, customer c, product p
  where  ((co.cust_po=:Customer_PO_Number and c.cust_name=:Customer_Name) or
         cd.order_no=:Order_Number) and cd.order_no=co.order_no and
         co.cust_id=c.cust_id and cd.product_id=p.product_id
order by cd.cust_line_no
```

Query Name: pal

```
select pc.order_no, pc.cust_line_no, (cd.cust_qty - sum(pd.qty_cases))
 from customer_order_d cd, pallet_card pc, pallet_d pd
  where pd.pallet_id=pc.pallet_id and pc.order_no=cd.order_no and
        pc.cust_line_no=cd.cust_line_no and pc.pallet_status='C'
group by pc.order_no, pc.cust_line_no, cd.cust_qty
```

Parent Query 1: cus   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
CUST_LINE_NO	CUST_LINE_NO	

Query Name: inv_nc

```
select pc.order_no, pc.cust_line_no, sum(pd.qty_cases), pd.production_date
 from pallet_card pc, pallet_d pd
  where pc.pallet_id=pd.pallet_id and pc.pallet_status='NC'
  group by pc.order_no, pc.cust_line_no, pd.production_date
order by pd.production_date
```

Parent Query 1: cus   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
CUST_LINE_NO	CUST_LINE_NO	

Query Name: inv

```
select pc.order_no, pc.cust_line_no, sum(pd.qty_cases), pd.production_date
 from pallet_card pc, pallet_d pd
  where pc.pallet_id  NOT IN (select sxr.pallet_id from ship_cross_ref sxr)
        and pc.pallet_id=pd.pallet_id and pc.pallet_status='C'
  group by pc.order_no, pc.cust_line_no, pd.production_date
order by pd.production_date
```

Parent Query 1: cus   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
CUST_LINE_NO	CUST_LINE_NO	

Query Name: shp   Query 5 of 5

```
select sxr.order_no, sxr.order_no_line, s.shipment_date, sum(pd.qty_cases)
 from   pallet_d pd, ship_cross_ref sxr, shipment s
  where  sxr.pallet_id=pd.pallet_id and sxr.shipment_no=s.shipment_no
  group by sxr.order_no, sxr.order_no_line, s.shipment_date
order by s.shipment_date
```

Parent Query 1: cus   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
ORDER_NO_LINE	CUST_LINE_NO	

Summary Settings 1 of 2

Summary Name	Field	Function	Type	Width	Format
INV_SUM	INVEN	Sum	NUM	6	
SHP_SUM	SHIP	Sum	NUM	6	
NC_SUM	NON_C	Sum	NUM	6	

Summary Settings 2 of 2

Summary Name	Print Group	Reset Croup
INV_SUM	G_opn	G_opn
SHP_SUM	G_opn	G_opn
NC_SUM	G_opn	G_opn

* SALES AND CONTRACT MANAGEMENT
** Current Order and Contract Management (A14)
*** Non - Compliant Production Lot Status (A144)

Report Name: ncsr2

Non Compliance Status Report

Contract No: 123456 Product ID: MIL-B-44059B
Line Item No: 1 Product Name: Beef Stew

Total No. of
Non-Compliance Units: 72

In Rework		Variance			
Production Date	Units	Production Date	Units	Date Requested	Status
02-MAR-91	36	03-MAR-91	36		

Query Settings

Query Name: con Query 1 of 4

```
select pc.order_no, pc.cust_line_no, pc.product_id, p.product_name,
        sum(pd.qty_cases)
  from pallet_card pc, product p, pallet_d pd
  where pc.order_no=:Contract_No and pc.product_id=p.product_id and
        pc.pallet_id=pd.pallet_id and
        (pc.pallet_status='RS' or pc.pallet_status='RR' or
        pc.pallet_status='RV')
group by pc.order_no, pc.cust_line_no, pc.product_id, p.product_name
```

Query Name: rej   Query 2 of 4

```
select pc.order_no, pc.cust_line_no, pc.product_id, pd.production_date,
        sum(pd.qty_cases)
  from pallet_card pc, pallet_d pd
  where pc.pallet_id=pd.pallet_id and pc.pallet_status='RS'
  group by pc.order_no, pc.cust_line_no, pc.product_id, pd.production_date
order by pd.production_date
```

Parent Query 1: con   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
CUST_LINE_NO	CUST_LINE_NO	

Query Name: rew   Query 3 of 4

```
select pc.order_no, pc.cust_line_no, pc.product_id, pd.production_date,
        sum(pd.qty_cases)
  from pallet_card pc, pallet_d pd
  where pc.pallet_id=pd.pallet_id and pc.pallet_status='RR'
  group by pc.order_no, pc.cust_line_no, pc.product_id, pd.production_date
order by pd.production_date
```

Parent Query 1: con   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
CUST_LINE_NO	CUST_LINE_NO	

Query Name: var   Query 4 of 4

```
select pc.order_no, pc.cust_line_no, pc.product_id, pd.production_date,
        sum(pd.qty_cases), pc.var_date, pc.var_status
  from pallet_card pc, pallet_d pd
  where pc.pallet_id=pd.pallet_id and pc.pallet_status='RV'
  group by pc.order_no, pc.cust_line_no, pc.product_id, pd.production_date,
           pc.var_date, pc.var_status
order by pd.production_date
```

Parent Query 1: con

Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	
CUST_LINE_NO	CUST_LINE_NO	

```
* FACTORY FLOOR CONTROL
  ** Factory Floor Scheduling
    *** Daily Production Schedule
        **** List of Daily Schedule

Report Name: daily_ord

                                 Daily Schedule

Schedule Id  101
Begin Period 01-JAN-91
End Period   22-JAN-91

  Order                                                             Start   #
  Number   Product                Machine                           Time   Hrs
  -------- ---------------------  ------------------------------  ------  ---

  100009-1   1026  CHICKEN BROTH   101   PUMP STATION # 1, 150 GAL           1
  100009-1   1026  CHICKEN BROTH   201   # 5 CAN FILLING LINE                1
  100009-1   1026  CHICKEN BROTH   301   STILL RETORTS                       1
  100011-1   1026  CHICKEN BROTH   101   PUMP STATION # 1, 150 GAL          80
  100011-1   1026  CHICKEN BROTH   201   # 5 CAN FILLING LINE                8
  100011-1   1026  CHICKEN BROTH   301   STILL RETORTS                       8

Query Name: main                                              Query  1 of 2

select schedule_id,begin_period,end_period
    from schedule_m
where schedule_id = :Schedule_Id_Number


Query Name: sub                                               Query  2 of 2

select sd.schedule_id, sd.order_no, sd.cust_line_no, cd.product_id,
           product_name, sd.machine_id, machine_desc, schedule_seq,
           commited_time
    from schedule_d sd, customer_order_d cd, product p, machine m
    where sd.order_no=cd.order_no
    and sd.cust_line_no=cd.cust_line_no
    and cd.product_id=p.product_id
and sd.machine_id=m.machine_id
```

Parent Query 1: main Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
SCHEDULE_ID	SCHEDULE_ID	

* MANUFACTURING PLANNING
** Contract Planning (A21)
*** Intermediate Demand Schedule (A21411)

Report Name: int_dmd1

Intermediate Demand Schedule

Contract Number: 123456
Customer ID : C200

Product ID	Effective Due Date	Required # Shifts
1026	31-MAY-91	10
1026	31-MAY-91	10
MIL-B-44059B	30-APR-91	10
MIL-O-44202A	30-APR-91	10

Query Settings

Query Name: main   Query 1 of 2

```
select order_no, cust_id
    from customer_order
where order_no = :Contract_Number
```

Query Name: data   Query 2 of 2

```
select order_no,product_id, effective_due_date,
           cust_request_date - effective_due_date
    from customer_order_d
order by product_id, effective_due_date
```

Parent Query 1: main   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
ORDER_NO	ORDER_NO	

* MANUFACTURING PLANNING
  ** Contract Planning (A21)
    *** Manufacturing Plan (A21412)

Report Name: mfg_pln1

Manufacturing Plan

Process Id 2001 FILL # 5 Avail Time 110

Month	Order No	Line No	Eff Due Date	No Shifts
31-MAY-91	123456	1	31-MAY-91	0.1
	123456	4	31-MAY-91	0.0
			Total for Month	0.1
31-AUG-91	100009	1	06-AUG-91	26.8
	100011	1	07-AUG-91	71.4
	100011	2	07-AUG-91	47.6
	100012	2	09-AUG-91	47.6
	100010	1	10-AUG-91	44.6

Query Settings

Query Name: main Query 1 of 2

```
select p.process_id, p.process_desc, sum(avail_time)
    from process p, machine_process mp, machine_avail ma
    where p.process_class = 'F'
    and p.process_id = mp.process_id
    and mp.machine_id = ma.machine_id
    and ma.period_length = 21
group by p.process_id, p.process_desc
```

Query Name: detail


```
select last_day(effective_due_date), order_no, cust_line_no,
           effective_due_date, process_id, (cust_qty/(avg(rate)*7))
    from customer_order_d cd, recipe_line rl, line_process lp
    where cd.product_id = rl.product_id
    and rl.production_line_id = lp.production_line_id
    and cd.cust_line_status <> 'CL'
group by process_id, effective_due_date, order_no, cust_line_no, cust_qty
```

Parent Query 1: main   Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
PROCESS_ID	PROCESS_ID	

Summary Settings 1 of 2

Summary Name	Field	Function	Type	Width	Format
month_sum	NO_SHIFTS	Sum	NUM	7	ZZZZ9.9

Summary Settings 2 of 2

Summary Name	Print Group	Reset Group
month_sum	G_mid	G_mid

* MANUFACTURING PLANNING
  ** Contract Planning (A21)
    *** Material Purchase Plan (A2142)

Report Name : mpp_mth1

Material Purchase Plan
based on orders with recipe's assigned

Last Day	Material Id	Material Desc	Total Monthly Qty	Units
31-MAY-91	61232	CHICKEN FAT	3.1	LBS
	61236	CHICKEN BROTH 8%	7.8	LBS
	71154	1/4" GROUND CARROT	2.8	LBS
	84443	RIBOTIDE	0.3	LBS
	85130	SALT, DRY OR LIQUID	2.0	LBS

Query Settings

Query Name: main   Query 1 of 1

```
select last_day(effective_due_date), ml.material_id, material_desc,
           material_uom, sum((cust_qty/amt_end_units)*amt_by_unit)
    from customer_order_d cd, recipe_master rm, material_list ml, material m
    where cd.cust_line_status <> 'CL'
    and cd.recipe_id = rm.recipe_id
    and cd.product_id = rm.product_id
    and rm.product_id = ml.product_id
and ml.material_id = m.material_id
```

```
* FACTORY FLOOR CONTROL
  ** Raw Material Control
     *** Material Requistion (A21425, A22133, A211)


Report Name: matl_pos


                             Material position report

Material 61311        CLAMS, QUAHOGS, 3/8" GROUND PL
Units LBS
Reorder Point
Reorder Qty

  Material Lot No  On Hand Qty  Status
  ---------------  -----------  ----------
                    ----------
  Total Quantity


Query Settings

Query Name: matl                                        Query  1 of 2

select material_id, material_desc, material_uom, reorder_point,
           reorder_qty
from material


Query Name: matl_lot                                    Query  2 of 2

select material_lot_no, material_id, vlot_on_hand, vlot_status
    from vendor_lot
 where vlot_on_hand > 0

Parent Query 1: matl                    Parent Query 2:

Child Columns            Parent 1 Columns          Parent 2 Columns

MATERIAL_ID               MATERIAL_ID
```

Summary Settings 1 of 2

Summary Name	Field	Function	Data Type	Data Width	Display Format
total_matl	VLOT_ON_HAND	Sum	NUM	10	

Summary Settings 2 of 2

Summary Name	Print Group	Reset Group
total_matl	G_matl	G_matl

* FACTORY FLOOR CONTROL
** Factory Floor Scheduling
*** Batch Sheet (A2222)

Report Name: batch1

Batch Sheet

Product: 1026 CHICKEN BROTH
Gallons 300.00
Target Cook Time: 999.0
Target Cook Temp: 999.00
Recipe Id 1

Material Id	Material Desc	Amount	Units
61232	CHICKEN FAT	40	LBS
61236	CHICKEN BROTH 8%	100	LBS
71154	1/4" GROUND CARROT	36	LBS
84443	RIBOTIDE	4	LBS
85130	SALT, DRY OR LIQUID	26	LBS

Query Settings

Query Name: recipe_m Query 1 of 2

```
select rm.product_id, product_name, gallons, target_cook_time,
           target_cook_temp, recipe_id
    from recipe_master rm, product p
    where rm.product_id = :product_identification
    and rm.recipe_id = :recipe_identification
and p.product_id=rm.product_id
```

Query Name: list

```
select ml.material_id, material_desc, unit_of_measure, amt_by_unit,
           r.product_id, r.recipe_id
    from recipe r, material_list ml, material m
    where r.material_id=ml.material_id
    and ml.material_id=m.material_id
and r.product_id=ml.product_id
```

Parent Query 1: recipe_m Parent Query 2:

Child Columns	Parent 1 Columns	Parent 2 Columns
PRODUCT_ID	PRODUCT_ID	
RECIPE_ID	RECIPE_ID	

Report Name: process_info

PRODUCT PROCESS INFORMATION

Product ID: 1026 Product Name: CHICKEN BROTH
Recipe ID: 1

Target Initial Temperature:	70	Degrees
Process Time:	22.3	Minutes
Process Temperature:	225	Degrees

Query Settings

Query Name: proc Query 1 of 1

```
select  rm.product_id, rm.recipe_id, p.product_name, rm.process_time,
          rm.process_temp, rm.target_init_temp
    from  recipe_master rm, product p
    where rm.product_id=:Product__ID and rm.recipe_id=:Recipe__ID and
rm.product_id=p.product_id
```

* FINISHED GOODS CONTROL
** Inventory Control
*** View Inventory Status

Report Name: fgood

Finished Goods Inventory Report

Product ID: 1026
Product Desc: CHICKEN BROTH

Production Date	Location	Label Id	Label Desc	Order No	Cust Line No	Number of Cases
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25
01-MAY-91	WAREHOUS	SHINE		100009	1	25

Query Settings

Query Name: inv   Query 1 of 1

```
select pc.product_id, p.product_name, pc.pallet_id, pc.location, pc.label_
          l.label_name, pc.order_no, pc.cust_line_no, sum(pd.qty_cases),
          pd.production_date
    from pallet_card pc, product p, pallet_d pd, label l
    where pc.pallet_id NOT IN (select sxr.pallet_id from ship_cross_ref sxr) a
         pc.product_id=p.product_id and pc.pallet_id = pd.pallet_id and
         pc.label_id=l.label_id (+)
group by pc.product_id, p.product_name, pc.pallet_id, pc.location,
```

Summary Settings 1 of 2

Summary Name	Field	Function	Type	Width	Format
SUM	QTY_CASES	Sum	NUM	9	

Summary Settings 2 of 2

Summary Name	Print Group	Reset Group
SUM	G_mil	G_mil

```
* FINISHED GOOD CONTROL
  ** Lot Tracking / Trace
     *** Lot Tracking (Raw -> Customer)
```

Report Name: rtrace

Customer Name	Production Date	Julian Date	Filling Line	Retort ID	Cook No.

Query Settings

Query Name: raw   Query 1 of 1

```
select C.cust_name, R.production_date, TO_CHAR(R.production_date, 'YDDD')
          R.filling_line, R.retort_id, R.cook_no
    from batch_d BD, retort R, customer C
    where BD.material_lot_no = :Material_Lot_Number
          and R.production_date=BD.production_date
          and R.filling_line=BD.filling_line
          and ROUND ((R.retort_start_hour+(R.retort_start_min/60) - .75),2) =R
and C.cust_id IN (select cust_id from pallet_card
```

* FINISHED GOODS CONTROL
  ** Lot Tracking / Trace
    *** Lot Tracking (Raw -> Finished Goods)

Report Name: ftrace

Product Id	Product name	Section
1026	CHICKEN BROTH	WAR
3011	RAVIOLI IN SAUCE	WAR
6111	WHITE CLAM SAUCE	WAR

Query Settings

Query Name: fin   Query 1 of 1

```
select distinct p.product_id, p.location, pp.product_name
    from pallet_card p, pallet_d pd, retort r, batch_d bd, product pp
    where bd.material_lot_no = :Material_Lot_Number
     and r.production_date=bd.production_date
     and r.filling_line=bd.filling_line
     and pp.product_id = p.product_id
     and ROUND((r.retort_start_hour+(r.retort_start_min/60) - .75),2) = ROUND(
and pd.cook_no=r.cook_no and pd.production_date=r.production_date
```

COMBAT RATION ADVANCED MANUFACTURING TECHNOLOGY DEMONSTRATION (CRAMTD)

Report on Quality Assurance Module Implementation
Part I

Mr. Richard Holowczak
Graduate School of Management
Rutgers, The State University of New Jersey

July 1994

1. Introduction

This report describes the development of the Quality Assurance portion of the CRAMTD Preliminary database. The raw material and finished goods testing procedures were implemented for Beef Chunks in Gravy, Tray Pack. The tested raw materials are the beef cubes and the tray pack containers. The finished goods tests are performed on completed tray packs. This report presents the Phase I module developed under STP #4. It will be followed by a report on the final database module developed under STP #16.

2. Raw Materials Inspection (Beef Cubes)

The testing procedures for beef cubes were developed under STP#3 and are described in the CRAMTD SPC Plan, Beef Chunks with Gravy, Half-Stew Table Trays, Approved on October 21, 1991. These consist of a series of Inspection Procedures (IPs) which are in turn made up of Laboratory Procedures (LPs) . There are a total of seven IPs for raw beef cubes and three IPs for tray packs. In this document, the layout of the oracle database form for each IP is given.

Raw Materials are sampled according to a Single Sampling Plan (SSP) , also outlined in the CRAMTD SPC Plan. Based on lot size, vendor inspection level and acceptable quality level (AQL) , the SSP determines the number of samples to be tested and the tolerance for defects from the incoming lot.

The storage and tracking of raw material inspections required adding several tables to the relational database. These included:

Lot Quality	An individual vendor's track record supplying a particular raw material.
Single Sample Plan	Lookup table for sample sizes based on lot size, vendor inspection level and AQL .
Test Procedure	Keeps track of inspection procedures, number of attributes and descriptions of each IP .
Test Attribute Spec	Keeps track of each attribute of an IP .
Material Test	Holds all of the raw material test results.
Lab Proc Detail	Holds all details and descriptions of LPs.

Using these tables, a series of forms were created as shown in this document. One electronic form for each IP was created along with forms for the SSP, Lot Quality History

by Vendor, Lot Quality History by Material and detailed lot quality history by vendor.

3. Finished Goods Inspection (Beef Cubes with Gravy, Tray Pack)

The testing procedure for finished goods were outlined in the CRAMTD SPC Plan. As with the raw material IPs, this document gave the Inspection Procedures, Lab Procedures and sample forms.

Finished goods are tested according to the Double Sampling Plan (DSP) . The DSP determines the number of finished goods samples to take based on lot size and number of prior defects.

Inspection procedure maintenance forms were also created to allow the creation and modification of IPs as well as association with raw materials or finished products.

4. Database Menu Structure

The menu structure for Raw Material Inspections and Finished Goods Inspections is as follows:

CRAMTD Applications Database Main Menu
- Factory Floor Control Menu
  - Quality Control Menu
    - Raw Material Inspection Procedures
      - * IPBCT0101 Color/Odor/Material Test
      - * IPBCT0102 Drain Weight Test
      - * IPBCT0103 Size of Beef Cubes
      - * IPBCT0104 Beef Cube Surface Fat > 1/8"
      - * IPBCT0105 Connective Tissue or Cartilage
      - * IPBCT0106 Size of Bones
      - * IPBCT0107 Microbiological Inspection
    - Raw Material Test Results
      - * Raw Material Test Results
      - * Material Lot History By Material
      - * Material Lot History By Vendor
    - Finished Goods Inspection Procedures
      - * IPBCT 2101 Net Weight Inspection
      - * IPBCT 2102 Foreign Material
      - * IPBCT 2102 Foreign Color
      - * IPBCT 2103 Foreign Odor and Flavor
      - * IPBCT 2104 Excessive Heating
      - * IPBCT 2105 Drain Weight of Beef
      - * IPBCT 2106 Gravy Consistency

Finished Goods Test Results

Test Maintenance and Customization Menu
* Single Sampling Plan
* Double Sampling Plan
* Material Inspection / Lab Procedures

Each IP form has been extensively customized for greater ease of use. Whenever possible, "Pop-up" menus were added to aid the user in looking up identification numbers. Some examples of this are the Material Id, Material Lot, Vendor Id and Employee Id pop up menus. On-line help screens were also programmed to allow the user on-line access to IP and LP information during the actual testing. Automatic totals and averages are calculated where applicable and the raw material acceptance decision cycle has been entirely automated based on pre-determined rules in the SSP and DSP.

5. Example Forms

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Test Maintenance and Customization Menu

                         Double Sampling Plan
+-----------------------------------------------------------------------+
|                                                                       |
|  Sequence        Lot Size                     Acceptance    Rejection |
|  Number      Low          High    Sample Size   Level        Level    |
|  1_____   0_______  - 3200_____   8_______    0________    2_______   |
|  2_____   0_______    3200_____   8_______    1________    2_______   |
|  ______   ________    _________   ________    _________    ________   |
|  ______   ________    _________   ________    _________    ________   |
|  ______   ________    _________   ________    _________    ________   |
|                                                                       |
+-----------------------------------------------------------------------+
|   Related Inspection Procedures                                       |
|   IPBCT2101_ Net_Weight_Inspection_________________________           |
|   IPBCT2102_ Foreign_Material______________________________           |
|   IPBCT2102_ Foreign_Color_________________________________           |
|   IPBCT2103_ Foreign_Odor_or_Flavor________________________           |
|   IPBCT2104_ Excessive_Heating_____________________________           |
|   IPBCT2105_ Drain_Weigh_of_Beef___________________________           |
|   IPBCT2106_ Gravy_Consistency_____________________________           |
|   IPBCT2107_ Viscosity_-_Brookfield_Method_________________           |
|   IPBCT2109_ Meat_Chunk_Size_______________________________           |
+-----------------------------------------------------------------------+
Type_the_sequence_number_for_this_sample._(1_if_first_sample,_2_if_second,_etc.)

* Factory Floor Control Menu
 ** Quality Control Menu

        +---------------------------------------------------+
        |      Finished Goods Inspection Procedure Menu     |
        +---------------------------------------------------+
        |  Product Id   Product Description                 |
        |  _________    ___________________________         |
        +---------------------------------------------------+
        |  Inspection Procedures                            |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        |  _________    ________________________________    |
        +---------------------------------------------------+
        |                                 Press Esc to exit |
        +---------------------------------------------------+
Type_a_Product_Id_(PRxxxx)_or_press_F9_for_a_listing.______________________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu

          +--------------------------------------------------------------+
          |      In-Process Material Inspection Procedure Menu          |
          +--------------------------------------------------------------+
          | Material Id  Material Description                           |
          | ___________  _______________________________                 |
          +--------------------------------------------------------------+
          | Inspection Procedures                                       |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          | __________   __________________________________________     |
          +--------------------------------------------------------------+
          |                                          Press Esc to exit  |
          +--------------------------------------------------------------+
Type_a_Material_Id_(MAxxxx)_or_press_F9_for_a_listing.__________________________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Test Maintenance and Customization Menu

                    Material Inspection/Lab Procedures
+--------------------------------------------------------------------------------+
|    Material Id: __________   Description: ________________________________   |
+--------------------------------------------------------------------------------+
|    Inspection Proc  AQL     Procedure Description                Attributes |
|    __________       _____%  ____________________________________  _____     |
|    __________       _____   ____________________________________  _____     |
|    __________       _____   ____________________________________  _____     |
|    __________       _____   ____________________________________  _____     |
|    __________       _____   ____________________________________  _____     |
|    __________       _____   ____________________________________  _____     |
|    __________       _____   ____________________________________  _____     |
|    __________       _____   ____________________________________  _____     |
|    __________       _____   ____________________________________  _____     |
|    __________       _____   ____________________________________  _____     |
+--------------------------------------------------------------------------------+
|    Lab Procedure  Seq #  Description                                         |
|    __________     _____  ____________________________________________        |
|    __________     _____  ____________________________________________        |
|    __________     _____  ____________________________________________        |
|    __________     _____  ____________________________________________        |
|    __________     _____  ____________________________________________        |
+--------------------------------------------------------------------------------+
Type_a_Material_Id_(MAxxxx)_or_press_F9_for_a_listing.__________________________
```

* Factory Floor Control Menu
** Quality Control Menu
*** Raw Material Test Results Menu

Material Lot History By Material

Material ID: MA51000___ Description: BEEF,_DICED_________________

Vendor Id	Vendor Name	Historical Acc	Historical Rej	Historical Pend	Historical Total	Current Acc	Current Rej	Inspection Level
V1000____	AAA_PROVISION_COMPANY_	1___	0___	0___	1___	1___	0___	Normal____
V1001____	FRESH_MEATS__________	0___	0___	1___	1___	0___	0___	Normal____

* Factory Floor Control Menu
** Quality Control Menu

Raw Material Inspection Procedure Menu

Material Id	Material Description

Inspection Procedures

Press Esc to exit

Type_a_Material_Id_(MAxxxx)_or_press_F9_for_a_listing.________________________

* Factory Floor Control Menu
** Quality Control Menu
*** Raw Material Test Results

Material Lot:	ML5000_____	Material:	MA51000___	BEEF,_DICED_______________	
Rec. Date:	12-DEC-1990	Vendor:	V1000_____	AAA_PROVISION_COMPANY_________	
Rec. Quant:	1200_______	Cur. Qty:	200_______	Status: A___	
Lots Accep.:	1__________	Lots Rej.:	0_________	Inspect. Level:	Normal________

Inspection Procedure		AQL	Samples Req	Samples Taken	Defects	Levels Acc	Levels Rej
IPBCT0101	Color/Odor/Material_Test____	.01__ %	20___	20___	0______	0___	1___
IPBCT0102	Drain_Weight_Test___________	1____	20___	20___	0______	0___	1___
IPBCT0103	Size_of_Beef_Cubes__________	1____	20___	20___	0______	0___	1___
IPBCT0104	Beef_Cube_Surface_Fat_>_1/8"	1____	20___	20___	0______	0___	1___
IPBCT0105	Connective_Tissue_or_Cartila	1____	20___	20___	0______	0___	1___
IPBCT0106	Size_of_Bones_______________	1____	20___	20___	0______	0___	1___
IPBCT0107	Microbiological_Inspection__	.01__	20___	20___	0______	0___	1___

Recommendation: ______________________________ Acc [_] Rej [_] Con [_]

* Factory Floor Control Menu
** Quality Control Menu
*** Test Maintenance and Customization Menu

Single Sampling Plan

Acceptable Quality Level: 1.0___ %

Lot Size Low	Lot Size High	Normal Ss	Normal Ac	Normal Re	Tightened Ss	Tightened Ac	Tightened Re	Reduced Ss	Reduced Ac	Reduced Re

Related Inspection Procedures

Type_the_Acceptable_Quality_Level_you_wish_to_view._(1.0%_or_0.01%_for_testing).

* Factory Floor Control Menu
** Quality Control Menu
*** Raw Material Test Results

Material Lot History by Vendor

Vendor Id	Vendor Name	Start Date	End Date
V1000_____	AAA_PROVISION_COMPANY_________	01-JAN-80_	12-AUG-92_

Material Lot	Vendor Lot	Material Id	Receive Date	Receive Qty	Status
ML5000_____	0200_______	MA51000____	12-DEC-1990_	1200_______	A_____

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Raw Material Inspection Procedures
+--------------------------------------------------------------------+-------------------+
|   Colo Odor Mate       Colo Odor Mate        Colo Odor Mate        | Foreign Color/Odor|
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | and Material Test |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | IP #:  IPBCT0101_ |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | Lot #:  ________  |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | AQL:    _____ %   |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | Date:  31-MAR-93_ |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | Inspect:________  |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | Samples:_____     |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | Status: _____     |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         +-------------------+
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         |    DATA ENTRY     |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         +-------------------+
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         |   Total Defects   |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         |                   |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | Color:    ______  |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         |                   |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | Odor:     ______  |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         |                   |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         | Material: ______  |
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         |          ---------|
|_ ____ ____ ____     _ ____ ____ ____     __ ____ ____ ____         |   Total:  ______  |
|To:____ ____ ____ Tot:____ ____ ____  Tot:____ ____ ____           +-------------------+
Type_a_Material_Lot_Number_(MLxxxx)_or_press_F9_for_a_listing.______________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Raw Material Inspection Procedures
+--------------------------------------------------------------+-------------------+
|   Wt.   %            Wt.   %              Wt.   %            |  Drain Weight of  |
|_ ____ ____        _ ____ ____         __ ____ ____           |     Beef Test     |
|_ ____ ____        _ ____ ____         __ ____ ____           | IP #:  IPBCT0102_ |
|_ ____ ____        _ ____ ____         __ ____ ____           | Lot #:  ________  |
|_ ____ ____        _ ____ ____         __ ____ ____           | AQL:    _____ %   |
|_ ____ ____        _ ____ ____         __ ____ ____           | Date:  31-MAR-93_ |
|_ ____ ____        _ ____ ____         __ ____ ____           | Inspect:________  |
|_ ____ ____        _ ____ ____         __ ____ ____           | Samples:_____     |
|_ ____ ____        _ ____ ____         __ ____ ____           | Status: _____     |
|_ ____ ____        _ ____ ____         __ ____ ____           +-------------------+
|_ ____ ____        _ ____ ____         __ ____ ____           |    DATA ENTRY     |
|_ ____ ____        _ ____ ____         __ ____ ____           +-------------------+
|_ ____ ____        _ ____ ____         __ ____ ____           | Total:    ______  |
|_ ____ ____        _ ____ ____         __ ____ ____           |                   |
|_ ____ ____        _ ____ ____         __ ____ ____           | Average:  ______  |
|_ ____ ____        _ ____ ____         __ ____ ____           |                   |
|_ ____ ____        _ ____ ____         __ ____ ____           | Largest:  ______  |
|_ ____ ____        _ ____ ____         __ ____ ____           |                   |
|_ ____ ____        _ ____ ____         __ ____ ____           | Smallest: ______  |
|_ ____ ____        _ ____ ____         __ ____ ____           |                   |
|_ ____ ____        _ ____ ____         __ ____ ____           | Range:    ______  |
|To:____       Total:____ ____     Total:____ ____             +-------------------+
Type_a_Material_Lot_Number_(MLxxxx)_or_press_F9_for_a_listing.______________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Raw Material Inspection Procedures
+--------------------------------------------------------------------+--------------------+
|  Wt.   %  Def.      Wt.   %  Def.      Wt.   %  Def.   | Size of Beef Cubes |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    |    Defect Test     |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | IP #:   IPBCT0103_ |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Lot #:   _______   |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | AQL:     _____ %   |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Date:   31-MAR-93_ |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Inspect:_______    |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Samples:_____      |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Status:  _____     |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    +--------------------+
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    |    DATA ENTRY      |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    +--------------------+
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Tot. Wt.:  ______  |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Reg. Wt.:  ______  |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    |                    |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Average:   ______  |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Largest:   ______  |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Smallest:  ______  |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Range:     ______  |
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    |--------------------|
|__ ___ ___ ___    __ ___ ___ ___    __ ___ ___ ___    | Total Def.: _____  |
|To:___      ___  Tot:___     ___  Tot:___      ___    +--------------------+
Type_a_Material_Lot_Number_(MLxxxx)_or_press_F9_for_a_listing.______________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Raw Material Inspection Procedures
+--------------------------------------------------------------------+--------------------+
|  >1/8 Def.          >1/8 Def.           >1/8 Def.       |  Surface Fat of    |
|__ ___ ___        __ ___ ___         __ ___ ___          |  Beef Cubes Test   |
|__ ___ ___        __ ___ ___         __ ___ ___          | IP #:   IPBCT0101_ |
|__ ___ ___        __ ___ ___         __ ___ ___          | Lot #:   _______   |
|__ ___ ___        __ ___ ___         __ ___ ___          | AQL:     _____ %   |
|__ ___ ___        __ ___ ___         __ ___ ___          | Date:   31-MAR-93_ |
|__ ___ ___        __ ___ ___         __ ___ ___          | Inspect:_______    |
|__ ___ ___        __ ___ ___         __ ___ ___          | Samples:_____      |
|__ ___ ___        __ ___ ___         __ ___ ___          | Status:  _____     |
|__ ___ ___        __ ___ ___         __ ___ ___          +--------------------+
|__ ___ ___        __ ___ ___         __ ___ ___          |    DATA ENTRY      |
|__ ___ ___        __ ___ ___         __ ___ ___          +--------------------+
|__ ___ ___        __ ___ ___         __ ___ ___          | Def. Cubes / Sample|
|__ ___ ___        __ ___ ___         __ ___ ___          | Average:   ______  |
|__ ___ ___        __ ___ ___         __ ___ ___          | Largest:   ______  |
|__ ___ ___        __ ___ ___         __ ___ ___          | Smallest:  ______  |
|__ ___ ___        __ ___ ___         __ ___ ___          | Range:     ______  |
|__ ___ ___        __ ___ ___         __ ___ ___          +--------------------+
|__ ___ ___        __ ___ ___         __ ___ ___          |      Totals        |
|__ ___ ___        __ ___ ___         __ ___ ___          | Def. Cubes:______  |
|__ ___ ___        __ ___ ___         __ ___ ___          | Def. Sampl:______  |
|To:___  ___    Total:___  ___    Total:___  ___          +--------------------+
Type_a_Material_Lot_Number_(MLxxxx)_or_press_F9_for_a_listing.______________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Raw Material Inspection Procedures

+-----------------------------------------------------------------------------+-------------------+
|    Wt.   %   Def.      Wt.   %   Def.      Wt.   %   Def.  | Connective Tissue/ |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               |  Cartilage Defect  |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | IP #:   IPBCT0105_ |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Lot #:  _________  |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | AQL:    ______ %   |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Date:   31-MAR-93_ |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Inspect:_________  |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Samples:_____      |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Status: _____      |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               +-------------------+
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               |     DATA ENTRY     |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               +-------------------+
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Tot. Wt.: _______  |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Reg. Wt.: _______  |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               |                    |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Average:  _______  |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Largest:  _______  |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Smallest: _______  |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Range:    _______  |
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               |-------------------|
|_ ___ ___ ___   _ ___ ___ ___   _ ___ ___ ___               | Total Def.: _____  |
|To:___     ____ Tot:____     ____ Tot:____     _ __         +-------------------+
Type_a_Material_Lot_Number_(MLxxxx)_or_press_F9_for_a_listing.________________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Raw Material Inspection Procedures

+------------------------------------------------------------+-------------------+
|   >0.3 Def.        >0.3 Def.        >0.3 Def.              |   Beef Bone Size   |
|_ ___ ___        _ ___ ___        _ ___ ___                 |    Defect Test     |
|_ ___ ___        _ ___ ___        _ ___ ___                 | IP #:   IPBCT0106_ |
|_ ___ ___        _ ___ ___        _ ___ ___                 | Lot #:  _________  |
|_ ___ ___        _ ___ ___        _ ___ ___                 | AQL:    ______ %   |
|_ ___ ___        _ ___ ___        _ ___ ___                 | Date:   31-MAR-93_ |
|_ ___ ___        _ ___ ___        _ ___ ___                 | Inspect:_________  |
|_ ___ ___        _ ___ ___        _ ___ ___                 | Samples:_____      |
|_ ___ ___        _ ___ ___        _ ___ ___                 | Status: _____      |
|_ ___ ___        _ ___ ___        _ ___ ___                 +-------------------+
|_ ___ ___        _ ___ ___        _ ___ ___                 |     DATA ENTRY     |
|_ ___ ___        _ ___ ___        _ ___ ___                 +-------------------+
|_ ___ ___        _ ___ ___        _ ___ ___                 | Def. Bones / Sample|
|_ ___ ___        _ ___ ___        _ ___ ___                 | Average:  _______  |
|_ ___ ___        _ ___ ___        _ ___ ___                 | Largest:  _______  |
|_ ___ ___        _ ___ ___        _ ___ ___                 | Smallest: _______  |
|_ ___ ___        _ ___ ___        _ ___ ___                 | Range:    _______  |
|_ ___ ___        _ ___ ___        _ ___ ___                 +-------------------+
|_ ___ ___        _ ___ ___        _ ___ ___                 |       Totals       |
|_ ___ ___        _ ___ ___        _ ___ ___                 | Def. Bones:______  |
|_ ___ ___        _ ___ ___        _ ___ ___                 | Def. Sampl:______  |
|To:___ ___    Total:___ ___    Total:___ ___                +-------------------+
Type_a_Material_Lot_Number_(MLxxxx)_or_press_F9_for_a_listing.________________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Raw Material Inspection Procedures

+-----------------------------------------------------------------------+--------------------+
| CFU/g Def.              CFU/g Def.              CFU/g Def.            |   Microbiological  |
|__ ____ ____           __ ____ ____            ____ ____ ____          |     Defect Test    |
|__ ____ ____           __ ____ ____            ____ ____ ____          | IP #:   IPBCT0107_ |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Lot #:   _________ |
|__ ____ ____           __ ____ ____            ____ ____ ____          | AQL:     ______ %  |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Date:   31-MAR-93_ |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Inspect:__________ |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Samples:______     |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Status:  ______    |
|__ ____ ____           __ ____ ____            ____ ____ ____          +--------------------+
|__ ____ ____           __ ____ ____            ____ ____ ____          |     DATA ENTRY     |
|__ ____ ____           __ ____ ____            ____ ____ ____          +--------------------+
|__ ____ ____           __ ____ ____            ____ ____ ____          |       CFU/g        |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Average:  ________ |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Largest:  ________ |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Smallest: ________ |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Range:    ________ |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Total:    ________ |
|__ ____ ____           __ ____ ____            ____ ____ ____          +--------------------+
|__ ____ ____           __ ____ ____            ____ ____ ____          |      Total         |
|__ ____ ____           __ ____ ____            ____ ____ ____          | Def. Sampl:_______ |
|To:____ ____      Total:____ ____       Total:____ ____                +--------------------+
Type_a_Material_Lot_Number_(MLxxxx)_or_press_F9_for_a_listing.________________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Raw Material Inspection Procedures

+-----------------------------------------------------------------------+--------------------+
|  Dimen      Dimen      Dimen      Dimen       Dimen       Dimen       | Tray Can Dimension |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         |     Defect Test    |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         | IP #:   IPBCT0008_ |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         | Lot #:  __________ |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         | AQL:     ______ %  |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         | Date:   31-MAR-93_ |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         | Inspect:__________ |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         | Samples:______     |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         | Status:  ______    |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         +--------------------+
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         |     DATA ENTRY     |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         +--------------------+
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         |   Total Defects    |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         |                    |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         | Dimension:________ |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         |                    |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         |         ---------- |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         | Total Def:________ |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         |                    |
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         | Pct. Def:  ______ %|
|__ ____   __ ____    __ ____    __ ____    ___ ____    __ ____         |                    |
|To:____      ____       ____       ____        ____        ____        +--------------------+
Type_a_Material_Lot_Number_(MLxxxx)_or_press_F9_for_a_listing.________________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Raw Material Inspection Procedures
+-----------------------------------------------------------+----------------------+
|  Dimen    Dimen    Dimen    Dimen    Dimen    Dimen       | Tray Flange Dimen-   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |  sion Defect Test    |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | IP #:   IPBCT0009_   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Lot #:  __________   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | AQL:    ______ %     |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Date:   31-MAR-93_   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Inspect:__________   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Samples:______       |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Status: ______       |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            +----------------------+
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |     DATA ENTRY       |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            +----------------------+
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |   Total Defects      |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |                      |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Dimension:________   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |                      |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |          ----------  |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Total Def:________   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |                      |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Pct. Def:  ______ %  |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |                      |
|To:____    ____    ____    ____    ____    ____            +----------------------+
Type_a_Material_Lot_Number_(MLxxxx)_or_press_F9_for_a_listing.________________
```

```
* Factory Floor Control Menu
 ** Quality Control Menu
  *** Raw Material Inspection Procedures
+-----------------------------------------------------------+----------------------+
|  Dimen    Dimen    Dimen    Dimen    Dimen    Dimen       |   Tray Can Dents     |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |     Defect Test      |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | IP #:   IPBCT0010_   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Lot #:  __________   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | AQL:    ______ %     |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Date:   31-MAR-93_   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Inspect:__________   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Samples:______       |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Status: ______       |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            +----------------------+
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |     DATA ENTRY       |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            +----------------------+
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |   Total Defects      |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |                      |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Can Dents:________   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |                      |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |          ----------  |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Total Def:________   |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |                      |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            | Pct. Def:  ______ %  |
|__ ____ __ ____ __ ____ __ ____ __ ____ __ ____            |                      |
|To:____    ____    ____    ____    ____    ____            +----------------------+
Type_a_Material_Lot_Number_(MLxxxx)_or_press_F9_for_a_listing.________________
```

COMBAT RATION ADVANCED MANUFACTURING TECHNOLOGY DEMONSTRATION (CRAMTD)

Report on Equipment Maintenance Module Implementation
Part I

Mr. A. Gangopadhyay
Department of Industrial Engineering
Rutgers, The State University of New Jersey

July 1994

1. Introduction

In a manufacturing plant producing combat rations or civilian packaged food products, management does not have the luxury of redundant process equipment. When equipment fails, it must be restored in a reason able period of time. Maintenance people need good documentation on preventive maintenance, repair procedures, and spare parts inventory. A major reason for a database and electronic work order forms is to document reasons for failure, identify repair procedure, document failed parts and maintain necessary spares.

The maintenance module of the CRAMTD Preliminary Database deals with general information about the production equipment and machinery, spare and repair parts, information on preventive and break-down maintenance of the equipment, and down time information of the production equipment. General information about equipment includes the equipment number, which is a unique identifier for each equipment, equipment description, manufacturer, vendor information, serial number, date in service, and warranty expiration date.

In the preventive maintenance schedule, information is kept for each machine on the tasks to be performed, the last date that each task was performed, and how often each task needs to be done. Each task has a unique identification number.

For break down maintenance, information is kept on the work order issued for the tasks to be done, the machine on which the maintenance task is done, the due date and date the task was requested, and who made the request. Information is also kept on when the task was completed, what parts were used for the repair, the date the parts were withdrawn from inventory, and the number of parts withdrawn. Also kept is information about the employee deployed for the tasks, and the dates and hours of work performed.

Each time a machine goes down for any reason, whether for scheduled maintenance or machine failure, we keep information about the cause of the down time, and the dates it went down and became serviceable again. The database also provides information about the minimum, maximum, and mean down time lengths for each machine.

This report presents the Phase I module developed under STP #4. It will be followed by a report on the final implemented module developed under STP #16.

2. Menus

The maintenance module consists of the following menu items.

1. Equipment Data Entry Form: This allows the operator to store new equipment to the database as well as retrieve information about existing equipment. This form is shown in Figure 1.

2. Preventive Maintenance History Form: This form serves two purposes. It is used to enter a preventive maintenance schedule for each machine, which includes a "task" and the "frequency" with which the task is to be done. Each time preventive maintenance is actually performed, this form is used to update the actual date that the maintenance was

done. This form is shown in Figure 2.

3. Work Order Data Entry Form: Every time a work order is issued, this form is used by the operator to enter the relevant information into the system. This form is shown in Figure 3.

4. Work Order History Form: For every work order, this is used to store information such as which tasks were performed, when the work was completed, what parts were withdrawn from inventory, and the dates and numbers of parts withdrawn, along with the employees that worked on it and the dates and hours they worked. This form is shown in Figure 4.

5. Downtime Description Form: "Down time" is time during which the equipment is not available for use. The reason could be a breakdown, time for preventive maintenance, or even for precautionary purposes. For each type of down time, this form is used to enter a description of the downtime. A unique downtime identifier is generated by the system which is used in other forms to refer to the type of downtime. This form is shown in Figure 5.

6. Machine Downtime Data Entry Form: Each time a machine goes down, this form is used to enter the downtime ID, which gives the reason for downtime (the system pops up a list of possibilities). This form is also used to enter the dates and times the machine went out of service, and the date and time the machine became available again. This form is shown in Figure 6.

7. Machine Downtime History: For each machine and each type of down time, the system calculates the maximum, minimum, and average down time lengths given any time period. This form is shown in Figure 7.

```
+-------------------------Equipment-Data-Entry--------------------------------+
|                                                                             |
|  Equipment No  __________        Equipment Name _______________             |
|                                                                             |
|                                                                             |
|                                                                             |
|  Manufacturer  __________        Vendor Id      ___________                 |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|  Serial Number __________        Date in Service ____________               |
|                                                                             |
|                                                                             |
|                                                                             |
|  Warranty Expiration Date ___________                                       |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
+-----------------------------------------------------------------------------+
-------------------------------------------------------------------------------
        Char Mode: Replace  Page 1                    Count: *0
```

Figure 1: Equipment Data Entry Form

```
+--------------======-Preventive-Maintenance-History-=====--------------------+
|                                                                             |
|                                                                             |
|           Equipment No  -----------                                         |
|                                                                             |
|                                                                             |
|      Task Number            Date Last Done              Frequency           |
|      -----------            --------------              ---------           |
|      -----------            --------------              ---------           |
|      -----------            --------------              ---------           |
|      -----------            --------------              ---------           |
|      -----------            --------------              ---------           |
|      -----------            --------------              ---------           |
|      -----------            --------------              ---------           |
|      -----------            --------------              ---------           |
|      -----------            --------------              ---------           |
|      -----------            --------------              ---------           |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
+-----------------------------------------------------------------------------+
+-----------------------------------------------------------------------------+

-------------------------------------------------------------------------------
        Char Mode: Replace  Page 1                      Count: *0
```

Figure 2: Preventive Maintenance History Form

```
+-----------------------Work-Order-Data-Entry------------------------------+
|                                                                          |
|  Work Order Id  ----------            Machine Id ----------              |
|                                                                          |
|  Due Date       ------------        Requested by -----------             |
|                                                                          |
|  Date Requested -------------            Task Id -----------             |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
+--------------------------------------------------------------------------+
----------------------------------------------------------------------------
        Char Mode: Replace  Page 1                    Count: *0
```

Figure 3: Work Order Data Entry Form

```
+----------------------=====-Work-Order-History=====---------------------------+
|                                                                              |
|Work Order Id              Date Completed            Task Id                  |
|------------               --------------            ----------               |
|                                                     ----------               |
|Date Requested             Machine Id                ----------               |
|-------------              ------------              ----------               |
|                                                                              |
|                      ===== Parts Consumed =====                              |
|                                                                              |
|             Material Lot   Date Withdrawn            Number Required         |
|             ------------   --------------            ---------------         |
|                                                                              |
|                                                                              |
|                        ===== Worker Hours =====                              |
|                                                                              |
|Employee Id                Production Date                Hours              |
|-----------                ---------------            ------------           |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
+------------------------------------------------------------------------------+
--------------------------------------------------------------------------------
        Char Mode: Replace  Page 1                    Count: *0
```

Figure 4: Work Order History Form

```
+--------------------------------------------------------------------+
|                      Down Time Description                         |
|                                                                    |
|          Downtime Id          Downtime Description                 |
|          ------------         ---------------------------         |
|                                                                    |
+--------------------------------------------------------------------+
--------------------------------------------------------------------
       Char Mode: Replace  Page 1                  Count: *0
```

Figure 5: Downtime Description

```
+----------------------Machine-Downtime-Data-Entry----------------------------+
|                                                                             |
|          Machine Id ______________                                          |
|                                                                             |
|                                                                             |
|                                    Start time                 End time      |
| Downtime Id       Start Date       HR : MN      End Date      HR : MN       |
| -----------       ----------       --    --     ---------     --    --      |
| -----------       ----------       --    --     ---------     --    --      |
| -----------       ----------       --    --     ---------     --    --      |
| -----------       ----------       --    --     ---------     --    --      |
| -----------       ----------       --    --     ---------     --    --      |
| -----------       ----------       --    --     ---------     --    --      |
| -----------       ----------       --    --     ---------     --    --      |
| -----------       ----------       --    --     ---------     --    --      |
| -----------       ----------       --    --     ---------     --    --      |
| -----------       ----------       --    --     ---------     --    --      |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
+-----------------------------------------------------------------------------+
-------------------------------------------------------------------------------
        Char Mode: Replace  Page 1                       Count: *0
```

Figure 6: Downtime Data Entry Form

```
+-------------------------=====-Machine-Downtime-History-=====-------------------------+
|                                                                                      |
|Machine Number _____________                    Start Date ____________               |
|                                                                                      |
|                                                End Date   ____________               |
|                                                                                      |
|                           Mean Down Time Length          Down Time Length Range      |
|   Down Time Id            Hour          Min                Low           High        |
|   ____________            --            --                 -- --         -- --       |
|   ____________            --            --                 -- --         -- --       |
|   ____________            --            --                 -- --         -- --       |
|   ____________            --            --                 -- --         -- --       |
|   ____________            --            --                 -- --         -- --       |
|   ____________            --            --                 -- --         -- --       |
|   ____________            --            --                 -- --         -- --       |
|   ____________            --            --                 -- --         -- --       |
|   ____________            --            --                 -- --         -- --       |
|   ____________            --            --                 -- --         -- --       |
|                                                                                      |
|                                                                                      |
|                                                                                      |
|                                                                                      |
+--------------------------------------------------------------------------------------+
----------------------------------------------------------------------------------------
          Char Mode: Replace  Page 1                        Count: *0
```

Figure 7: Downtime History Form

3. Database Relations:

The relations created for the maintenance module are shown below:

Name	Null?	Type
DOWN_TIME_ID	NOT NULL	CHAR(10)
DOWN_TIME_DESC		CHAR(40)

Down_time_detail

Name	Null?	Type
MACHINE_ID	NOT NULL	CHAR(10)
DOWN_TIME_ID	NOT NULL	CHAR(10)
START_DATE		DATE
START_HR		NUMBER
START_MIN		NUMBER
END_DATE		DATE
END_HR		NUMBER
END_MIN		NUMBER

Machine

Name	Null?	Type
MACHINE_ID	NOT NULL	CHAR(10)
MACHINE_LOCATION		CHAR(10)
MACHINE_DESC		CHAR(40)
STD_SHIFT_LENGTH		NUMBER
SERIAL_NO		CHAR(30)
VENDOR_ID		CHAR(10)
MANUFACTURER		CHAR(30)
DATE_IN_SERVICE		DATE
WARRANTY_EXP_DATE		DATE

Work_order

Name	Null?	Type
WORK_ORDER_ID	NOT NULL	CHAR(10)
MACHINE_ID		CHAR(10)
WO_DUE_DATE		DATE
REQUESTED_BY		CHAR(10)
WORK_ORDER_TYPE		CHAR(20)

DATE_REQUESTED		DATE
DATE_COMPLETED		DATE

Wo_parts_detail

Name	Null?	Type
WORK_ORDER_ID	NOT NULL	CHAR(10)
TASK_ID	NOT NULL	CHAR(10)
MATERIAL_LOT_NO	NOT NULL	CHAR(10)
NUMBER_REQUIRED		NUMBER
DATE_WITHDRAWN		DATE

Wo_task_emp_detail

Name	Null?	Type
WORK_ORDER_ID	NOT NULL	CHAR(10)
TASK_ID	NOT NULL	CHAR(10)
EMPLOYEE_ID	NOT NULL	CHAR(10)
HOURS		NUMBER
PRODUCTION_DATE		DATE

Pm_history

Name	Null?	Type
MACHINE_ID	NOT NULL	CHAR(10)
TASK_ID	NOT NULL	CHAR(10)
DATE_LAST_DONE		DATE
FREQUENCY		NUMBER

Task

Name	Null?	Type
TASK_ID	NOT NULL	CHAR(10)
TASK_DESCRIPTION		CHAR(40)

Wo_task_xref

Name	Null?	Type
WORK_ORDER_ID	NOT NULL	CHAR(10)
TASK_ID	NOT NULL	CHAR(10)

4. IDEF1X Model

The part of the IDEF1X model that is relevant to the maintenance module is shown in Figure 8. It consists of nine entities and a total of 31 attributes.

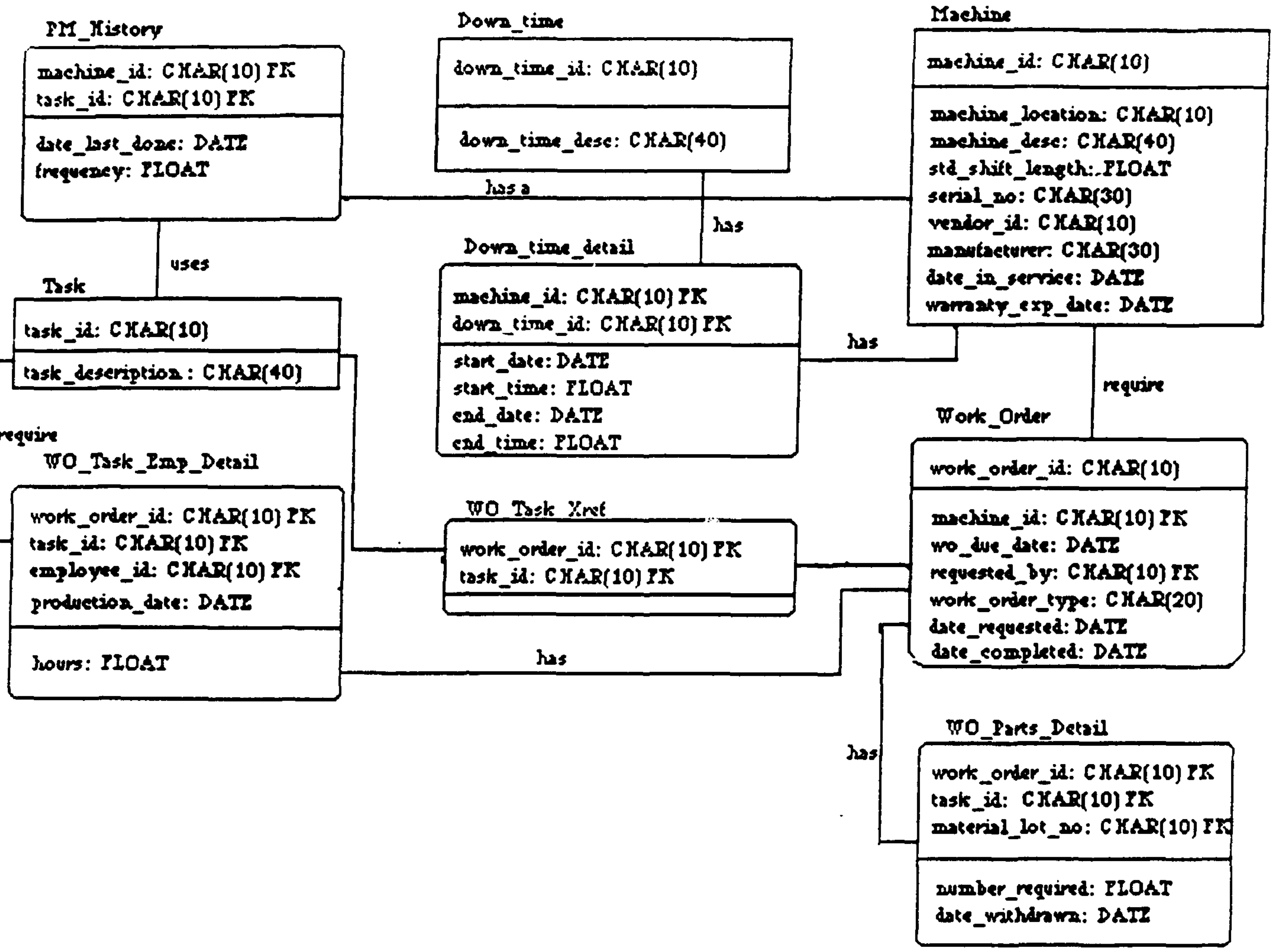


Figure 8: IDEF1X for Maintenance Module